



# **AJONEUVOJEN TIEDONSIIRTOMENETELMÄT JA DIAGNOOSILAITTEET**

Mikko Juvonen

Opinnäytetyö  
Toukokuu 2014  
Auto- ja kuljetustekniikka  
Auto- ja korjaamotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Auto- ja kuljetustekniikan koulutusohjelma  
Auto- ja korjaamotekniikan suuntautumisvaihtoehto

JUVONEN, MIKKO:  
Ajoneuvojen tiedonsiirtomenetelmät ja diagnoosilaitteet

Opinnäytetyö 57 sivua  
Toukokuu 2014

---

Sähköisten järjestelmien lisääntyminen ajoneuvoissa on kasvattanut haasteita niiden väliselle tiedonsiirrolle. Järjestelmien kehittymisen myötä myös vianetsintä on monimutkaistunut. Tämän työn tarkoituksena oli perehtyä ajoneuvoissa käytettäviin tiedonsiirtomenetelmiin ja tutkia vianetsintää ammattilaiselle sekä tavalliselle kuluttajalle suunnatulla laitteistolla. Tutkimus suoritettiin vertailemalla diagnoosilaitteita.

Tutkimus osoitti, että kuluttajalle suunnatun diagnoosilaitteen ominaisuudet ovat rajalliset ja sen hyöty vianetsinnässä vähäinen. Vertailupohjana käytettiin ammattimaista diagnoosilaitetta. Tutkimuksessa selvisi, mitä ominaisuuksia diagnoosilaitteelta tarvitaan.

Tekniikan kehittyessä omatoiminen vianetsintä on vaikeutunut. Tästä huolimatta, halpojen diagnoosilaitteiden määrä markkinoilla on lisääntynyt. Tutkimuksessa todettiin, että suuntaus on väärä ja vianetsintä olisi syytä jättää ammattilaisille.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Automobile and Transport Engineering  
Option of Automobile and Garage Engineering

JUVONEN, MIKKO:  
Vehicles Data Transfer Methods and Diagnostic Equipments

Bachelor's thesis 57 pages  
May 2014

---

The increasing of electronic systems in vehicles has created challenges for the data transfer between them. Along the development of systems, the troubleshooting has become more complex. The purpose of this thesis was to explore data transfer methods in vehicles and examine troubleshooting with professional, as well as average consumer directed equipments. The examination was conducted by comparing diagnostic equipments.

The examination indicates that a consumer directed equipment has limited features and it's benefit in troubleshooting is minor. A professional equipment was used as a baseline. The examination reveals what kind of features are required from a diagnostic equipment.

As technology develops, an independent troubleshooting becomes more difficult. Despite this, the number of inexpensive diagnostic equipment on the market has increased. The examination states that the trend is wrong and the troubleshooting should be done by specialists.

---

Key words: data transfer methods, diagnostic equipment, troubleshooting.

## SISÄLLYS

1	JOHDANTO .....	6
2	TIEDONSIIRRON PERUSTEET AJONEUVOSSA .....	7
2.1	Verkkotopologia .....	7
2.1.1	Väylätologia .....	7
2.1.2	Tähtitologia.....	8
2.1.3	Rengastologia .....	8
2.1.4	Verkkotologia .....	9
2.2	Väylän käyttämisen periaatteet .....	10
2.2.1	Keskitetty järjestelmä.....	10
2.2.2	Osittain hajautettu järjestelmä .....	10
2.2.3	Kokonaan hajautettu järjestelmä .....	11
2.2.4	Determinisesti säädelty järjestelmä .....	12
2.3	Viestin lähettämisen periaatteet .....	12
2.4	Ajoneuvoväylien vaatimukset ja jaottelu .....	13
3	CAN-VÄYLÄ.....	15
3.1	Perusteet .....	15
3.2	CAN ja OSI-malli .....	16
3.3	Fyysinen kerros.....	17
3.4	Siirtoyhteyserros .....	19
3.5	CAN-väylän diagnostiikka .....	22
4	LIN-VÄYLÄ.....	24
4.1	Perusteet .....	24
4.2	Tiedon välittäminen .....	24
4.3	Viestin rakenne .....	25
5	FLEXRAY .....	27
5.1	Perusteet .....	27
5.2	Fyysinen kerros.....	28
5.3	Siirtoyhteyserros .....	29
6	MOST-VÄYLÄ .....	33
6.1	Perusteet .....	33
6.2	Laitteisto ja tiedonsiirto.....	34
7	ON-BOARD DIAGNOSTICS.....	37
7.1	Järjestelmän tarkoitus ja kehittyminen.....	37
7.2	OBD-protokollat ja diagnoosiliitäntä.....	38
7.3	Järjestelmän vikakoodit ja toiminnot .....	39
8	TESTILAITTEVERTAILU .....	42

8.1 Ammattimainen testilaite.....	42
8.2 Kuluttajalle suunnattu testilaite .....	46
9 POHDINTA .....	52
LÄHTEET .....	55

## 1 JOHDANTO

Ajoneuvojen kehittyessä elektronisten järjestelmien ja komponenttien määrä lisääntyy jatkuvasti. Elektroniikan osuuden kasvaessa aiheutuu kustannus-, paino-, tila- ja asennusongelmia. Aiemmin ajoneuvojen sähköiset järjestelmät rakennettiin itsenäisiksi, mutta kehitys on pakottanut erilliset järjestelmät yhtenäistymään keskenään ja asettanut uusia vaatimuksia. Sähköjärjestelmän on oltava kustannustehokas, järjestelmällinen ja helposti laajennettava. Tämän vuoksi on kannattavaa liittää kaikki erilliset järjestelmät samaan yhtenäiseen ketjuun eli väylään. Väylällä kulkeva tieto on yhteistä ja se tavoittaa kaikki sitä tarvitsevat. Tällä ratkaisulla saadaan selkeä ja tehokkaasti toimiva verkko.

Tämän opinnäytetyön tarkoituksena on tutkia nykyaikaisten ajoneuvojen tiedonsiirtomenetelmiä. Työssä selvitetään, millä tavalla tietoa välitetään ajoneuvossa ja minkälaisia ratkaisuja erilaiset käyttökohteet ajoneuvoväyliltä vaativat. Tavoitteena on koota kattava kuvaus tämän päivän sähköisen verkon ominaisuuksista ja toiminnoista. Erilaisten väylätyyppien ohella paneudutaan myös tietoon, joka on tietyiltä osin tallennettava autojen ohjainlaitteista. Kyse on auton sisäisestä valvontajärjestelmästä, joka kulkee nimellä On-Board Diagnostics. Tämä järjestelmä tallentaa auton pakokaasupäästöihin vaikuttavat viat ja sen vikamuisti tarkastetaan määräaikaikatsastuksessa testiliitynnän kautta, joka on osa ajoneuvon tietoväylää. Lopuksi suoritetaan vertailu autojen sähköisten järjestelmien diagnoosilaitteista ja pohditaan vianhakuun vaadittavia ominaisuuksia sekä ammattilaisen laitteistolla, että tavalliselle kuluttajalle suunnatulla laitteistolla.

Tämä työ vastaa siis seuraaviin kysymyksiin: millä tavalla tietoa välitetään, mitä tietoa valvotaan ja minkälaisella laitteilla tietoa diagnosoidaan.

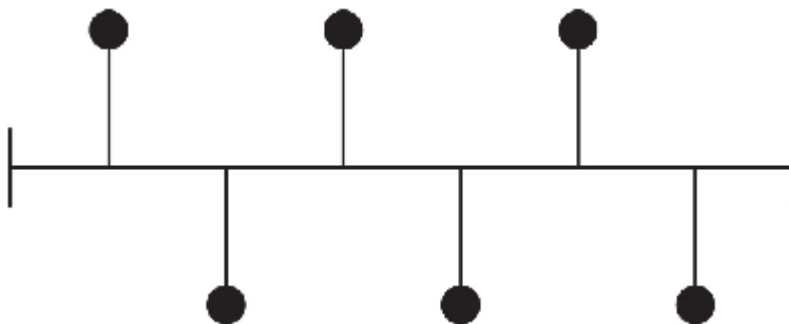
## 2 TIEDONSIIRRON PERUSTEET AJONEUVOSSA

### 2.1 Verkkotopologia

Tiedonsiirrossa verkkotopologialla tarkoitetaan tapaa, jolla laitteet on yhdistetty toisiinsa. Nimitys verkko syntyy, jos havainnollistetaan tietoa siirtäviä elementtejä solmuilla ja tiedonsiirtovälinettä viivoilla. Solmuja tiedonsiirrossa voidaan kutsua myös verkon tilaajiksi tai asemiksi. Verkkotopologia käsittää siis usean solmun sekä niiden välisen tiedonvälittämismenetelmän eli väylän. Jokaisen solmun on oltava yhteydessä vähintään yhteen toisista solmuista, jotta se voi osallistua tiedon siirtämiseen. Verkkotopologia määrittelee verkolle tietynlaisia ominaisuuksia, joten eri tarkoituksiin käytetään erilaista verkkoa. Kaikki verkkotopologiat perustuvat neljään perustopologiaan, mutta erilaisia rakenteita saadaan yhdistelemällä näitä topologioita. (Juhala 2008, 4–5.)

#### 2.1.1 Väylätopologia

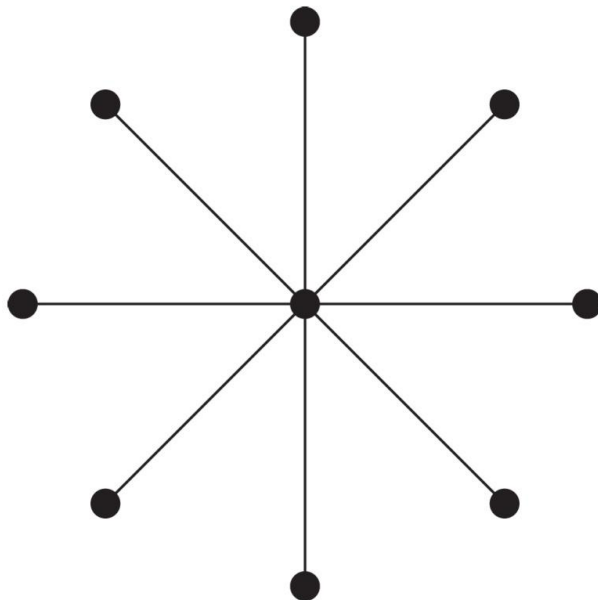
Ensimmäinen perustopologia on väylätopologia, jota kutsutaan myös lineaariseksi väyläksi. Kuviosta 1 nähdään sen perusidea, joka on yksi johdin, johon kaikki solmut ovat liittyneet (McQuerry 2008, 3). Tämän kaltaiseen verkkoon on helppo lisätä uusia tilaajia, eikä sen toiminta pääty, vaikka jokin solmuista menisi epäkuuntoon. Toisaalta taas pääjohtimen vioittuessa koko verkon toiminta päättyy. (Juhala 2008, 5.)



KUVIO 1. Väylätopologian rakenne (McQuerry 2008, 3)

### 2.1.2 Tähtitopologia

Toinen perustopologia on tähtitopologia. Siinä kaikki solmut ovat kuvion 2 mukaan liittyneet yhteen keskussolmuun, joka voi olla aktiivinen tai passiivinen (McQuerry 2008, 3). Ero aktiivisen ja passiivisen keskussolmun välillä on se, että aktiivinen keskussolmu käsittelee ja jakaa tietoa. Passiivinen keskussolmu vain liittää muut solmut yhteen. Tähtitopologiaan on myös helppo lisätä solmuja, jos vain vapaita liittimiä on jäljellä. Aikaisempaan topologiaan viitaten, myöskään tämä topologia ei lopeta toimintaansa yhden solmun vioituttua, mutta keskussolmun meneminen epäkuntoon lopettaa koko verkon toiminnan. Toiminta loppuu, oli kyseessä sitten aktiivinen tai passiivinen keskussolmu. Verkon toimintavarmuutta voidaan lisätä käyttämällä useita keskussolmuja, joihin tärkeät solmut liittyvät rinnakkaisesti. (Juhala 2008, 5–6.)



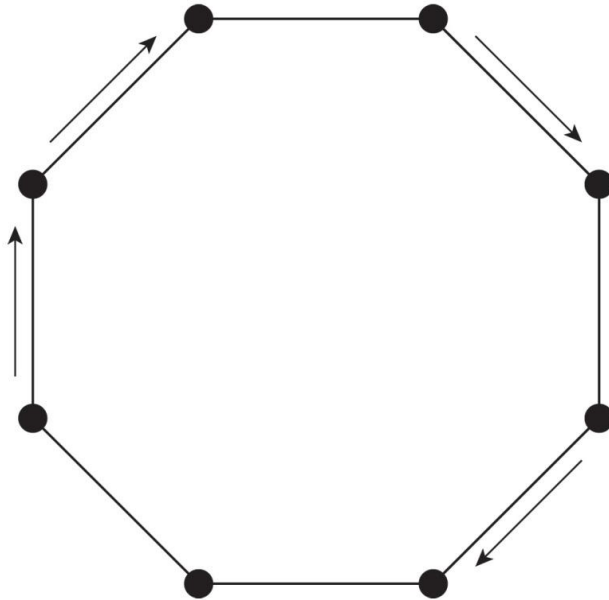
KUVIO 2. Tähtitopologian rakenne (McQuerry 2008, 3)

### 2.1.3 Rengastopologia

Kolmas perustopologia on rengastopologia. Rengastopologiassa nimensä mukaisesti, solmut ovat liittyneet aina kahteen vierekkäiseen solmuun muodostaen renkaan. Kuviosta 3 havaitaan perusperiaate (McQuerry 2008, 3). Renkaassa tiedonsiirto voi olla yhteen suuntaan tai molempiin suuntiin. Yhteen suuntaan kiertäessä tieto liikkuu aina samaan suuntaan. Tällaisen topologian ongelma on, että kun yksi solmu rikkoutuu, koko verkko lakkaa toimimasta. Mikäli tieto liikkuu kahteen suuntaan, on verkko rakennettu



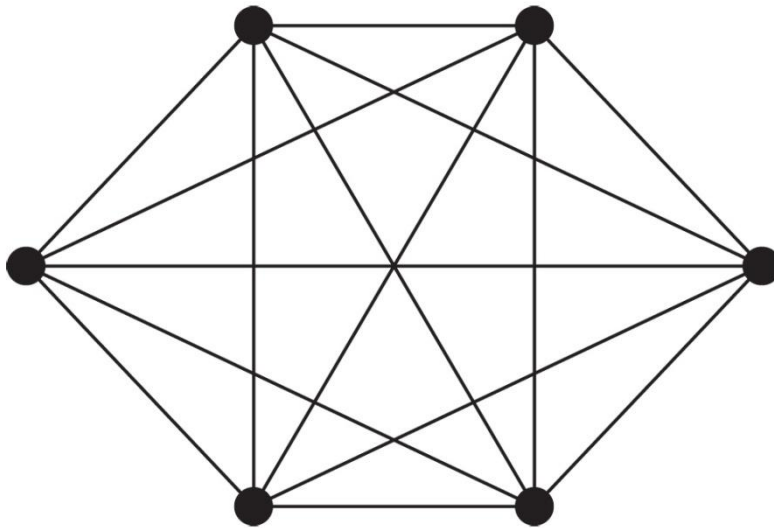
kaksinkertaiseksi. Kaksinkertaisessa verkossa on kaksi rengasta sisäkkäin, joista molemmat renkaat kuljettavat tietoa eri suuntiin. Renkaat ovat yhdistyneet solmukohdista ja näin ollen yhden solmun rikkoutuessa verkko jatkaa toimintaansa kiertäen viallisen solmun. Useamman solmun hajoaminen kuitenkin lopettaa verkon toiminnan. (Juhala 2008, 6; McQuerry 2008, 3.)



KUVIO 3. Yksinkertaisen rengastopologian rakenne (McQuerry 2008, 3)

#### 2.1.4 Verkkotopologia

Neljäs perustopologia on verkkotopologia. Siinä jokainen solmu liittyy yhteen tai useampaan muuhun solmuun. Silloin kun jokainen solmu on liittynyt joka ikiseen toiseen solmuun kuvion 4 mukaisesti, puhutaan täydellisesti verkkomaisesta verkosta (McQuerry 2008, 3). Tämän tyyppinen verkko on erittäin vakaa, koska solmun tai liittynnän vioittuessa eri reittivaihtoehtoja on monia. Toisaalta taas tällainen verkko on kallista valmistaa. Täydellisesti verkottunut verkko on väylämäinen viestin lähetyksen osalta, koska jokainen solmu vastaanottaa kaikki lähetykset. Tiedonsiirron osalta sillä on tähtitopologian ominaisuus kiertää ongelmakohta liityntähäiriön sattuessa. (Juhala 2008, 7.)



KUVIO 4. Täydellisen verkkotopologian rakenne (McQuerry 2008, 3)

## 2.2 Väylän käyttämisen periaatteet

Verkkotopologian lisäksi väyläjärjestelmiä voidaan jakaa myös tiedonkäsittelyn kannalta. Viestiä lähetettäessä väylä täytyy saada haltuun, mutta se, ketkä saavat edes pyrkiä väylälle ja milloin, määräytyy järjestelmän mukaan. Puhutaan keskitetyistä järjestelmistä, osittain hajautetuista tai täysin hajautetuista järjestelmistä. Näiden lisäksi voidaan käyttää ennustettavaa eli determinististä haltuunottoa.

### 2.2.1 Keskitetty järjestelmä

Näistä ensimmäisenä mainittu keskitetty järjestelmä on lähes samanlainen kuin vanha releohjattu järjestelmä, joten se on käytössä vain pienissä järjestelmissä. Keskitetyssä järjestelmässä on vain yksi ohjain, jossa on sisääntulot ja ulostulot, mutta erona releohjattuun järjestelmään on tietokoneyksikkö jolle ohjelmoidaan kaikki toiminnot. (Juhala 2008, 8–9; Myllymäki 2013, 31).

### 2.2.2 Osittain hajautettu järjestelmä

Huomattavasti yleisempi järjestelmä on osittain hajautettu järjestelmä, jossa on yksi ohjainyksikkö ja sisään- sekä ulostulomoduuleja, jotka on yleensä sijoitettu

toimilaitteiden lähelle. Järjestelmässä siis yksi solmuista toimii ohjainlaitteena sekä muut solmut ovat sisään- ja ulostulomoduuleja. Ohjainlaitteena toimivaa solmua sanotaan isännäksi muiden ollessa orjia. Isäntäsolmu toteuttaa kaikki ohjausfunktiot käyttäen orjasolmuja vain tiedon keräämiseen ja välittämiseen. Tieto siis tulee antureilta ja kytkimiltä sisään- ja ulostulomoduuliin, josta se välitetään ohjainlaitteelle. Ohjainlaite eli isäntäsolmu käsittelee tiedon ja lähettää oikeanlaisen käskyn sille moduulille eli orjalle, johon oikea toimilaite on kytketty. Näin isäntä- ja orjaperiaate toteutuu ja väylää käytetään vain pelkkänä anturi- ja toimilaitteväylänä eli kenttäväylänä. Orjasolmuja voidaan vaihtaa keskenään tai kokonaan uusiin, koska toimintaa määräävä ohjelma on ladattu isäntäsolmuun. (Myllymäki 2013, 33–36.)

### 2.2.3 Kokonaan hajautettu järjestelmä

Yllä mainitun lisäksi todella yleinen järjestelmä on kokonaan hajautettu järjestelmä, jossa kaikki siihen liittyneet solmut ovat ohjainlaitteita eli isäntäsolmuja. Kaikki solmut ovat samanarvoisia ja ne ovat yhdistetty toisiinsa väyläkaapelilla. Jokainen solmu käsittelee anturi- ja kytkintietoja ja jakaa sitä tietoa tarvittaessa muiden solmujen kanssa. Tässä järjestelmässä väylä toimii prosessoriväylänä, koska sitä käytetään tietojen vaihtamiseen solmujen välillä globaalina ohjausfunktion toteuttamiseksi. (Myllymäki 2013, 37–39.) Usean isännän järjestelmässä kaikki tilaajat ovat samanarvoisia ja pystyvät tavoittelemaan väylälle pääsyä ilman muiden apua. Tästä seuraa usein törmäystilanteita, mikäli useat eri solmut yrittävät lähettää viestiä väylälle samanaikaisesti. Samanlaista ongelmaa ei synny isäntä- ja orjaperiaatteella toimivissa järjestelmissä, missä isäntä pyytää tietoja ja orja vain vastaa pyyntöön. Törmäystilanteiden välttämiseksi käytetään esimerkiksi tärkeysjärjestystä. Jokaiselle solmulle on määrätty tärkeysjärjestys ja tärkeän solmun lähettäessä vähemmän tärkeät eivät pääse läpi. Kun väylä on taas vapaa eikä tärkeämpi solmu lähetä viestiä, jonossa olleet pääsevät läpi. Usean isännän järjestelmästä on myös hyötyä, koska kun mikään solmuista ei toimi järjestelmän säätelijänä, ei yhden solmun vioittuminen kaada verkkoa. (Juhala 2008, 9.)

### 2.2.4 Deterministisesti säädelty järjestelmä

Näiden yllä mainittujen käytössä olevien järjestelmien lisäksi voidaan väylälle pääsyä säädellä deterministisesti eli ennustettavasti. Tässä järjestelmässä poikkeava piirre on, että väylälle pääsy on ennalta määrätty. Valmiin aikataulun mukaan jokaisella solmulla on tietty aikaikkuna, minkä aikana se voi lähettää viestejä väylälle. Aikataulu voi olla kiinteä tai muuttuva mutta kuitenkin ennakoon määriteltä. Verkon tilaajien sisäisten kellojen täytyy toimia erittäin tarkasti synkronoituna, jotta lähetysikkunoita noudatetaan tarkasti. Yleensä tällaisessa järjestelmässä ei käytetä pääsolmua. (Juhala 2008, 9.)

## 2.3 Viestin lähettämisen periaatteet

Väylän haltuunoton yhteydessä on hyvä miettiä, mikä saa viestit liikkumaan väylässä. Milloin on tarpeellista lähettää viesti ja miten se löytää oikealle vastaanottajalle. Viestin lähettäminen tapahtuu kahdesta syystä, joko jokin tapahtuma laukaisee lähettämisen tarpeen tai ennalta määrätyn aikataulun mukaan on aika lähettää viesti. Viestin lähdettyä väylälle on tärkeää, että se päätyy oikealle vastaanottajalle, joka osaa hyödyntää sanoman. On viestejä, joita kaikki verkon tilaajat saattavat tarvita, mutta on myös paljon tietoa, jota vain tietty tilaaja tarvitsee. Siksi on hyvä osoittaa viesti vain sitä tarvitseville. Se tapahtuu lisäämällä viestille suora kohdeosoite tai merkitsemällä viesti tietyllä tunnisteella. (Juhala 2008, 8, 12–14.)

Viestin lähettäminen tapahtuu siis tietystä syystä. Ensimmäistä kutsutaan tapahtumaohjaukseksi. Siinä jokin tapahtuma käynnistää tarpeen lähettää viesti. Esimerkkinä vaikkapa halutaan avata ikkuna ja painetaan napista. Tällöin napin painaminen käynnistää viestin lähettämisen tarpeen, koska ikkunan katkaisijalta on lähetettävä pyyntö lasinnostimelle laskea ikkuna. Toinen esimerkki voisi olla saapuva anturitieto, joka edellyttää toimenpiteitä. Tapahtumia voi siis olla useampi samaan aikaan ja moni tilaaja yrittää ottaa väylää haltuun samaan aikaan. Näin pääsee tapahtumaan, koska asemat eivät jokaisessa tapauksessa ole tahdistettuja toisiinsa nähden. Kuitenkin vain yksi solmu voi käyttää kerralla väylää ja törmäysten välttämiseksi on olemassa useampia eri keinoja. Usean lähettäjän pyrkiminen väylälle saattaa ylikuormittaa verkkoa, mikä voi muodostua suureksi ongelmaksi. Toisaalta taas

satunnaisesti esiintyvät tapahtumat parhaimmassa tapauksessa pääsevät todella nopeasti väylälle verrattuna aikaperustaisiin järjestelmiin. (Juhala 2008, 12–13.)

Toinen käytössä oleva säätömekanismi on aikaohjattu järjestelmä. Tässä järjestelmässä verkossa on ennalta määrätty aikataulu, jossa lähetykset käsitellään verkon suunnittelussa määritellyllä tavalla. Jokaisella solmulla on tietty aikaikkuna, minkä aikana se voi lähettää viestin. Kun aikataulu on suoritettu, alkaa kierros alusta ja sama lähetyssykli toistuu. Tällainen järjestelmä on hyvin tarkka ja luotettava, koska mikäli jokin tilaaja ei lähetä omassa aikaikkunassaan viestiä, vaikka näin pitäisi, todetaan se rikkiinäksi ja se pudotetaan pois verkosta. Tämä tapahtuu sen takia, ettei viallinen solmu häiritsisi verkon toimintaa. Ennalta määriteltä aikataulu mahdollistaa nopean tiedonsiirron, mutta satunnaiset lähetykset eivät aina mahdu tiukan aikataulun väliin ja joutuvat joissain tilanteissa suuren viiveen uhriksi. Aikaperustaiselle järjestelmälle on asetettu kovat vaatimukset, koska sen käytöllä pyritään korvaamaan mekaaninen yhteys esimerkiksi ohjauksessa. Tämän takia viestien on saavuttava ajallaan, tärkeiden viestien viiveen eli latenssiajan on oltava erittäin lyhyt, eikä minkäänlainen vikatilanne saa tehdä järjestelmästä täysin toimimatonta. (Juhala 2008, 13–14.)

Lähetetyn viestin perille löytäminen varmistetaan oikeanlaisella osoitteenmuodostuksella. Se voi olla tilaajaperustainen, viestiperustainen tai lähetysperustainen. Ensimmäisessä menetelmässä viestillä on kohdeosoite ja jokainen solmu vertaa vastaako viestin osoite omaa osoitetta. Jos osoite täsmää, tilaaja ottaa viestin vastaan. Toisessa menetelmässä itse viestille annetaan tunniste. Verkon tilaajat määrittelevät, tarvitsevatko juuri tuollaisella tunnisteella liikkuvaa viestiä. Näin siis viestin lähettäjän ei tarvitse tietää viestin vastaanottajasta mitään. Lähetysperustainen osoitteenmuodostus tarkoittaa, että jokaisella viestillä on tietty aikaikkuna, jonka sisällä viesti lähetetään, ja viesti tunnistetaan tämän perusteella. Tämä menetelmä on yleensä yhdistetty kahteen aikaisemmin mainittuun järjestelmään. (Juhala 2008, 8.)

## **2.4 Ajoneuvoväylien vaatimukset ja jaottelu**

Ajoneuvossa väylätekniikan käyttämiseen asetetaan tiettyjä vaatimuksia. Ajoneuvojen järjestelmät käyttävät erilaisia väyliä, koska eri tarkoituksiin soveltuu tietynlainen väylätyyppi. Esimerkiksi tiedonsiirtonopeus jakaa väylät useisiin luokkiin. Jokaisessa

käyttökohteessa ei välttämättä tarvita juuri nopeinta väylää. Nopeuden lisäksi väylän luotettavuus ja reaaliaikaisuus merkitsevät. Mikäli mietitään turvallisuuteen liittyviä järjestelmiä, esimerkiksi lukkiutumattomia jarruja, tarvitaan niissä ensiluokkaista häiriön suodattamista ja virheiden havainnointia. Samaa vaatimustasoa ei vaadita vaikkapa mukavuuteen liittyviltä järjestelmiltä. Lisäksi järjestelmän täytyy reagoida täysin eri nopeudella kuin esimerkiksi ikkunannostimen moottorin, jossa pieni viive ei aiheuta vaaratilannetta. (Juhala 2008, 17–18.) Taulukossa 1 on ajoneuvoväylät jaoteltu eri luokkiin Society of Automotive Engineers eli SAE:n mukaan (Koivisto, Mikkolainen & Rantala 2012, 183; Virtanen, 11).

TAULUKKO 1. Ajoneuvoväylien jaottelu (Koivisto, Mikkolainen & Rantala 2012, 183; Virtanen, 11)

	Siirtonopeus	Käyttö	Väylä
A-luokka	Alle 10 kBit/s	Anturi- ja kytkintieto	LIN
B-luokka	Alle 125 kBit/s	Sisätila	Hidas-CAN
C-luokka	1 MBit/s asti	Voimalinja	Nopea-CAN
Turvallisuuskriittiset reaaliaikajärjestelmät	10 MBit/s asti	Alusta	Flexray
Viihdelaitteet	24,8 MBit/s	Telematiikka	MOST

Nykypäivän ajoneuvossa saatetaan käyttää yllä mainitun taulukon jokaista eri väylätyyppiä sekä eri protokollia. Ne eivät suoraan keskustele keskenään vaan niiden välille tarvitaan gateway eli yhdyskäytävä. Yhdyskäytävä muuttaa väylältä saadun tiedon sopivaan muotoon, jotta toinen väylätyyppi myös ymmärtää viestin. Se ei muokkaa viestin sisältöä, vaan tekee siitä ymmärrettävän. Väylällä voi olla useita hajautettuja yhdyskäytäviä, jotka ovat yhteydessä toisiinsa. Eli siis jokaiselle väylätyypille on oma yhdyskäytävänsä, joka muuttaa tiedon muille sopivaksi. Toisena vaihtoehtona on käyttää vain yhtä keskusyhdyskäytävää. (Koivisto ym. 2012, 183.)

### 3 CAN-VÄYLÄ

#### 3.1 Perusteet

CAN eli Controller Area Network on ajoneuvojen hajautettujen ohjausjärjestelmien reaaliaikaiseen tiedonsiirtoon suunniteltu tiedonsiirtoväylä. Se on autojen lisäksi laajalti käytössä myös muussa teollisuudessa, kuten esimerkiksi työkoneissa ja hisseissä. CAN soveltuu hyvin moneen koneeseen, mikäli välitettävät sanomat ja tiedonsiirtomatka ovat lyhyitä. Massatiedonsiirtoon käytetään erityyppisiä väyläratkaisuja. CAN on enemmän prosessoriverkko kuin kenttäväylä, vaikka jossain kohteissa sitä käytetään kenttäväylän tapaan. (Alanen & Scholliers 2004, 81.) CAN-väylän alkujuuret ylettyvät vuoteen 1983, jolloin Robert Bosch GmbH käynnisti sen kehittämisen. Vuonna 1986 Bosch esitteli virallisen CAN-protokollan ja vuonna 1992 sarjamuotoista CAN-väylää käytettiin ensimmäistä kertaa autossa, joka oli merkiltään Mercedes-Benz. (CAN in Automation.)

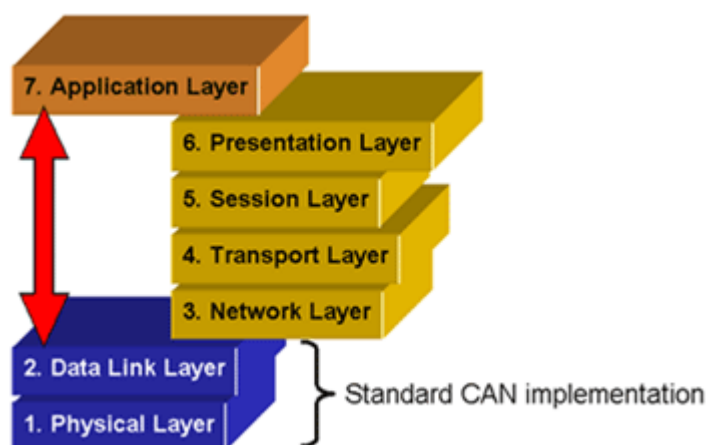
CAN-väylän tarkoituksena on vähentää johdotuksen määrää, saada sama tieto helpommin kaikille sitä tarvitseville sekä tehdä diagnosoinnista helpompaa. Kokonaisuudessaan sähköjärjestelmästä tulee selkeä ja modulaarinen, ja siihen on helppo lisätä uusia laitteita. Käyttökohteesta riippuen CAN-väylää käytetään kahdella eri nopeudella, jotka on standardoitu International Organization Engineering eli ISO:n ja SAE:n mukaan. Hidas-CAN on standardoitu ISO-11898-3 standardin mukaan ja se käsittää siirtonopeuden 125 kBit/s saakka. Nopea-CAN on standardoitu ISO-19898-2 standardin mukaan ja käsittää siirtonopeuden 1 MBit/s asti. Protokollaltaan samanlaiset standardit eroavat vain väylällä käytettävän jännitetasen osalta. SAE:n standardi J2284:n kolme kohtaa käsittää siirtonopeudet 125 kBit/s, 250 kBit/s ja 500 kBit/s saakka. Hitaamman väylän käyttökohteina ovat mukavuuspuolelle painottuvat toiminnot, vaikkapa ilmastoinnin säätäminen. Nopeammalla väylällä hoidetaan voimalinjaan liittyvät toiminnot, kuten moottorin säätäminen. (Alanen & Scholliers 2004, 82, 84–85.)

CAN-väylä on tyypiltään yleisimmin lineaarinen väylä ja sitä kutsutaan usean isännän väyläksi, jossa jokainen solmu voi yrittää itsenäisesti lähettää väylälle viestin. Solmujen määrälle ei ole loogista ylärajaa, mutta käytännössä rajan määrää lähetin-vastaanotinpiiri. Solmujen määrä on noin 200 kpl enimmillään kun käytetään erilliskomponentteja.

Väylän maksimipituus taas määräytyy käytetyn nopeuden mukaan, joka on 1 MBit/s nopeudella 40 m ja 50 kBit/s nopeudella jopa 1000 m. CAN-väylän viestinnässä viestejä ei lähetä lähettäjän eikä vastaanottajan osoitteella, vaan yleisesti vastaanotettavaksi, ja jokainen viestiä tarvitseva voi sen vastaanottaa. Viestissä käytetään tunnistetta, jonka mukaan jokainen solmu päättää, tarvitseeko viestin sisältöä. Lineaarisen väylätopologian ansiosta yhden solmun vioittuminen ei kaada koko verkkoa. (Alanen & Scholliers 2004, 85–86.)

### 3.2 CAN ja OSI-malli

Tietoliikenteessä yleisesti eritasoiset toiminnot järjestetään kerrostetusti. Näin on myös CAN-väylässä, jossa elektroniikka ja ohjelmisto on jaettu eri kerroksille (Juhala 2008, 34). ISO OSI-malli eli Open System Interconnection Model muodostaa havainnollisen mallin ja perustan tiedonsiirtoprotokollien kuvailulle ja vertailulle. OSI-mallissa tiedonvälityksen eri järjestelmät on kuvattu kerroksina, joten monimutkaiset tehtävät on järjestelty selkeisiin osa-alueisiin. Yksinkertaisessa tiedonvälittämisessä ei kaikki kerroksia tarvita, kuten ei tarvita CAN-väylässäkään. (Juhala 2008, 10.) Kuviosta 5 havaitaan CAN-väylän käyttämät kerrokset, jotka ovat sovelluskerroksen (application layer) lisäksi OSI-mallin kaksi alinta kerrosta, eli fyysinen kerros (physical layer) ja siirtoyhteyskerros (data link layer) (Shandle 2003).

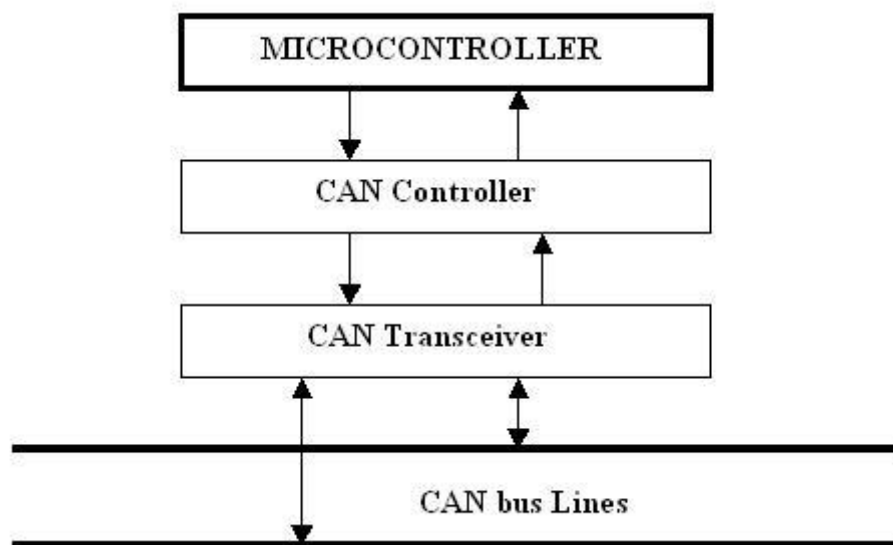


KUVIO 5. CAN-väylän käyttämät kerrokset OSI-mallista (Shandle 2003)



### 3.3 Fyysinen kerros

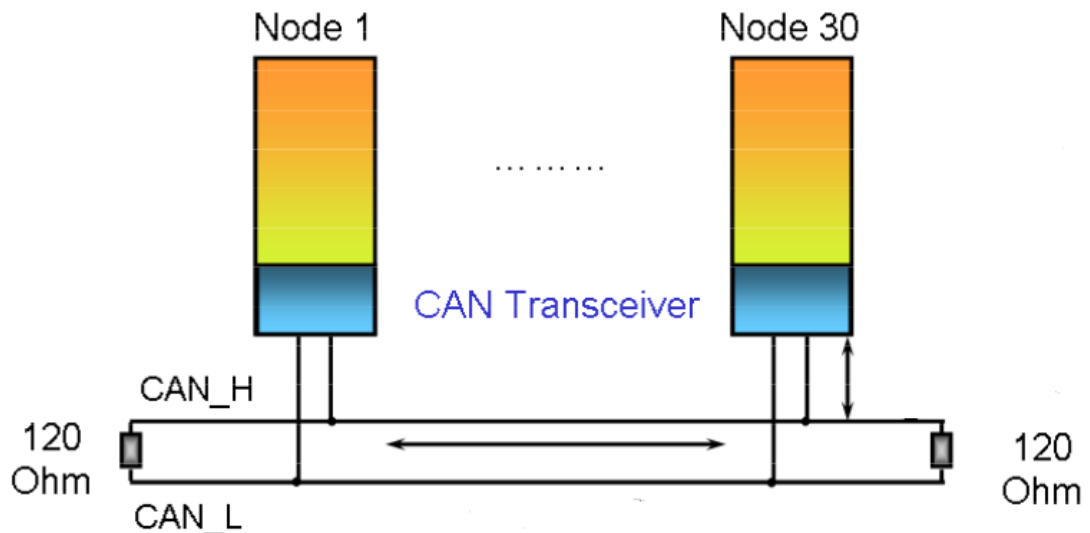
CAN-väylän fyysiseksi kerrokseksi kutsutaan laitteita, jotka siirtävät tietoa eri ohjainyksiköiden välillä. Fyysinen kerros käsittää väylän kaapeloinnin, lähetin-vastaanotin-yksikön, väyläohjaimen sekä mikro-ohjainlaitteen. Käytännössä nämä kaikki muodostavat ohjainlaitteen eli verkon solmun. Kuten aikaisemmin kuviosta 4 huomattiin, fyysinen kerros on alin kerros CAN-väylässä. (Koivisto ym. 2012, 182.) CAN-väylän fyysinen kerros on standardoitu ISO:n 11898 standardissa (Alanen & Scholliers 2004, 84). Kuviossa 6 nähdään CAN-solmun periaatekuva, jossa ylimpänä oleva mikro-ohjainlaite on yhteydessä antureihin sekä toimilaitteisiin (CAN interface in embedded system). Sen tehtävä on siis ottaa antureilta saatu tieto vastaan ja välittää se väyläohjaimelle sekä myös ohjata toimilaitteita käskyn saatuaan. Väyläohjaimen tehtävä on toimia tulkkina mikro-ohjaimen ja lähettimen välillä. Se muuntaa tietovirtaa siirtokelpoiseen muotoon ja se vastaa lähettämisestä sekä vastaanottamisesta. Alimpana näkyvän lähettimen tehtävä on vahvistaa väyläohjaimelta saatu signaali, tuottaa vaadittu jännitetaso ja lähettää käsitelty bittivirta väylälle. (Juhala 2008, 31.)



KUVIO 6. CAN-solmu (CAN interface in embedded systems)

CAN-väylän lineaarisen topologian vuoksi väyläkaapeli kulkee jokaisen aseman kautta ja se päätetään päätevastuksilla. CAN-väylän kahdesta kaapelista käytetään nimityksiä CAN-H ja CAN-L. Päätevastukset ovat suuruudeltaan 120 ohmia. ISO:n CAN-standardeissa määritetään käytettäväksi parikaapelia, ja korkeilla nopeuksilla yleensä käytetäänkin kierrettyä parikaapelia, suurimmilla nopeuksilla jopa suojattua.

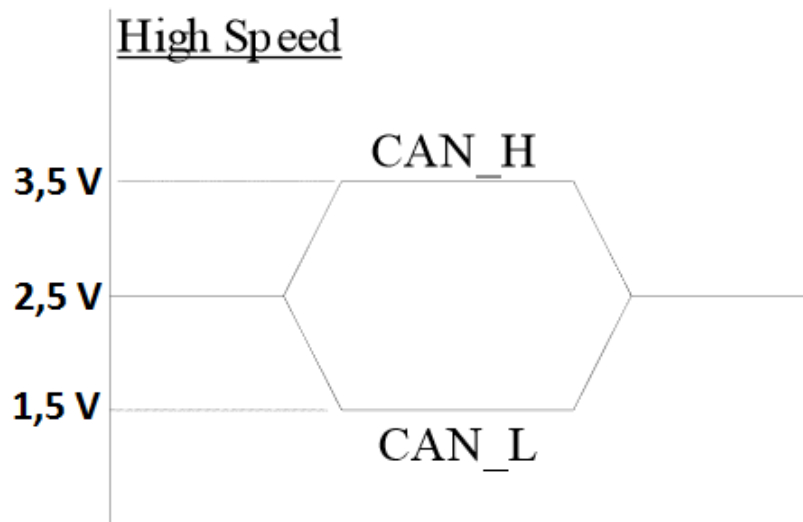
Pienemmillä nopeuksilla riittää jopa suora parikaapeli. Liitintyyppiä ei ole standardissa määritetty, mutta se yleensä tehdään vasta sovelluskohtaisissa standardoinneissa. (Alanen & Scholliers 2004, 86–87.) Kuvio 7 havaitaan peruseriaate väylästä (Virtanen, 17 muokattu).



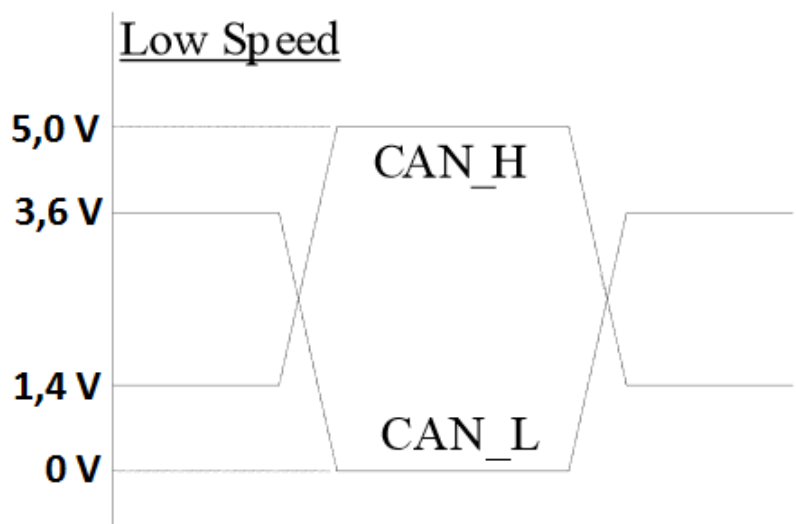
KUVIO 7. CAN-väylän periaatekuva (Virtanen, 17 muokattu)

CAN-ohjainlaitteen tehtävä on lähettää ja vastaanottaa loogisia tiloja 0 ja 1, joita voidaan kutsua biteiksi. Väylällä nämä tilat välitetään jännitetiloina. CAN-lähetin-vastaanotin muokkaa loogiset tilat jännitetasoiksi väylälle sekä toisinpäin. Väylältä tulevat jännitetasot muokataan loogisiksi tiloiksi. Näistä tiloista käytetään nimityksiä hallitseva ja väistävä, jossa 0 on hallitseva bitti.

Nopeassa CAN-väylässä eli esimerkiksi voimalinjaan tarkoitetussa käytetään väistävällä tilassa 2,5 V jännitettä molemmissa johtimissa, hallitsevassa tilassa CAN-L-johtimessa 1,5 V jännitettä ja CAN-H-johtimessa 3,5 V jännitettä. Hitaassa CAN-väylässä eli esimerkiksi mukavuusvarusteisiin kuuluvassa väylässä käytetään väistävällä tilassa CAN-L 0 V ja CAN-H 5 V jännitteitä. Hallitsevassa tilassa jännitteet ovat CAN-L 3,6 V ja CAN-H 1,4 V. (Koivisto ym. 2012, 185.) Kuvio 8 havainnollistaa jännitetasoja nopeassa CAN-väylässä, alkaen väistävästä tilasta, ja kuvio 9 kuvaa jännitetasoja hitaassa CAN-väylässä, alkaen hallitsevasta tilasta (Virtanen, 19 muokattu). CAN-väylällä tiedonsiirrossa käytetään NRZ eli Non-Return to Zero koodausmenettelyä, jonka mukaan bitin päättyessä ei vaadita, että kahden peräkkäisen ja samanarvoisen bitin välissä tarvitsisi välttämättä palata takaisin nollatasolle (Juhala 2008, 31).



KUVIO 8. Nopean CAN-väylän jännitetasot (Virtanen, 19 muokattu)



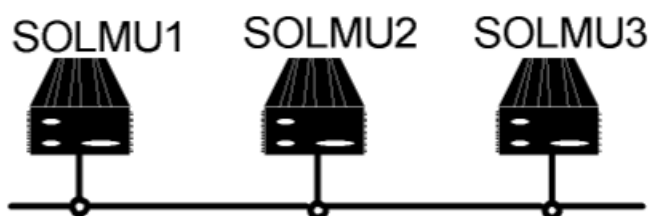
KUVIO 9. Hitaan CAN-väylän jännitetasot (Virtanen, 19 muokattu)

### 3.4 Siirtoyhteyskerros

Siirtoyhteyskerros on standardoitu ISO:n standardissa 11898-1 ja se pitää sisällään väylän kehyksen tyypin, rakenteen ja sen, miten niitä käytetään väylän sisäisessä tietojärjestelmässä (Myllymäki 2013, 45). CAN-väylässä käytetään aikaisemmin mainittua viestiperustaista osoitteenmuodostusta, eli viestillä ei varsinaisesti ole osoitetta vaan tunnistenumero. Jokainen verkon solmu määrittelee itse, tarvitseeko juuri sellaisella tunnisteella tulevaa viestiä. Solmuilla on siis tieto, mitä kukin tunniste tarkoittaa. Tunnisteen pituus vaihtelee hieman. Standardiformaatissa tunniste on 11 bittiä ja laajennetussa se on 18 bittiä. (Alanen & Scholliers 2004, 87.)

Jokainen solmu voi pyrkiä lähettämään viestin väylälle, mikäli se on vapaana. Jos syntyy tilanne, että useampi solmu yrittää lähettää samaan aikaan, tilanne ratkaistaan tunnistenumeron mukaan. Se viesti, jolla on korkein prioriteetti eli alhaisin tunnisteiden binääriarvo, saa väylän haltuun ilman tiedon häviämistä tai viivettä. Jokainen solmu lähettää omaa viestiään väylälle ja vertailu tapahtuu bitti bitiltä. Kun solmu yrittää lähettää väistävää bittiä, mutta väylällä on sillä hetkellä hallitseva bitti, häviää solmu kilpailun ja lopettaa viestin lähetyksen. Lähetys uusitaan väylän ollessa taas vapaa. Kuvio 10 havainnollistaa kilpailua väylälle pääsemisessä (Virtanen, 22). Ilman tämän kaltaista kilpailua väylälle pääsyssä tapahtuisi törmäyksiä, jotka aiheuttaisivat virheitä viesteihin. (Juhala 2008, 35–36.)

Kolme solmua yrittää lähettää sanoman yhtäaikaan



SOLMU 1: 0 0 0 1 0 1 1 0 0 0 1 = 177 (esim. moottorin kierrosnop.)  
 SOLMU 2: 0 1\*1 0 0 1 0 1 1 0 1 = 813 (esim. jäähdytysveden lämpöt.)  
 SOLMU 3: 0 0 0 1 1 0 1 1\*0 1 0 = 218 (esim. ajoneuvon nopeus)

---

VÄYLÄ : 0 0 0 1 0 1 1 0 0 0 1 = 177 (moottorin kierrosnopeus)

\* Asema huomaa menettäneensä valtuuden väylään, lopettaa lähettämisen ja jatkaa sanoman vastaanottajana. "Moottorin kierrosnopeus" voittaa valtuuden.

KUVIO 10. Kilpailu väylälle pääsystä (Virtanen, 22)

CAN-protokolla käsittää neljä erilaista viestimuotoa eli kehystä. Ensimmäinen on sanomakehys, joka nimensä mukaisesti sisältää tietoa. Toinen on kyselykehys, jolla jokin solmu voi kertoa tarvitsevansa tiettyä tietoa. Kolmas on virhekehys, jossa ilmoitetaan mahdollisesta virheestä tai viasta. Viimeisenä on ylikuormakehys, millä vastaanottava solmu voi pyytää lisää aikaa juuri vastaanottamaansa viestin käsittelyyn. (Alanen & Scholliers 2004, 87–88.) Kuviossa 11 on esitetty CAN-viestin kehysrakenne (Myllymäki 2013, 47).



KUVIO 11. CAN-viestin kehysrakenne (Myllymäki 2013, 47)

Kuvioon 11 viitaten CAN-viestin kehysrakenne muodostuu seuraavista osista:

- Viestin aloitus (Start of Frame) hallitsevalla bitillä, jonka tarkoitus myös myöhempien täytebittien kanssa synkronoida solmujen toimintaa.
- Viestin tunniste (Identifier), joka määrittelee viestin sanoman sekä prioriteetin. Standardimuodossa 11 bittiä ja laajennetussa 18 bittiä.
- Ohjausbitti (Remote Transmission Request), joka yhdessä viestin tunnisteiden kanssa muodostaa haltuunottokentän. Ohjausbitti määrittää, onko kyseessä sanoma- vai kyselykehys.
- IDE-bitti (Identifier Extension Bit) kertoo, onko kyseessä normaalimuotoinen vai laajennettu viesti.
- r0 = varattu, r1 = ei käytössä.
- Tietokentän pituuden (Data Length Code) määrittävä osuus.
- Tietokenttä (DATA), joka käsittää viestin tietosisällön. Samassa viestissä voidaan myös lähettää useita sanomia.
- Tarkastusluku (Cyclic Redundancy Checksum), jonka avulla voidaan laskennallisesti havaita mahdollisia virheitä tiedonsiirrossa.
- Kuittauskenttä (ACK), jota ei määritä lähettäjä vaan viestin vastaanottaja.
- Viestin lopetus (End of Frame) seitsemällä väistyvällä bitillä.
- Kehysten väli (Interframe Space) eli kolme väistyvää bitti, jotka yhdessä viestin lopetuksen kanssa muodostavat kymmenen peräkkäistä väistyvää bittiä. Näiden jälkeen muut solmut voivat aloittaa viestin lähettämisen. (Alanen & Scholliers 2004, 88; Juhala 2008, 37–38.)

CAN-väylän johtimiin saattaa kulkeutua häiriöitä ja muutenkin virheitä on mahdollista esiintyä. Siksi CAN-protokollassa on määritelty keinoja suorittaa virheentarkastusta. Lähettävä solmu voi tarkastaa, näkyykö väylällä juuri se bitti, minkä on kirjoittanut, tai onko saanut kuittausbitin asianmukaisesti. Vastaanottavat solmut voivat vertailla tarkastussummaa (CRK) tai rikkoa tahallaan bit-stuffing sääntöä. Bit-stuffing sääntö tarkoittaa, että viiden samanlaisen bitin jälkeen pitää lisätä vastakkainen bitti. Näiden

edellä mainittujen keinojen lisäksi on kaikkien verkon solmujen tehtävä suorittaa kehystarkastusta. Solmut siis tarkkailevat väylällä kulkevien viestien kehysrakenteiden niitä osia, jotka pysyvät viestin muodosta riippumatta aina samanlaisina. Poikkeavuudet kertovat virheiden tapahtumisesta. Jos virheitä havaitaan, kaikki solmut hylkäävät kyseisen viestin ja sen lähettäminen aloitetaan alusta. Mikäli virheiden määrä kasvaa tietyllä solmulla, se menee passiiviseen virhetilaan, jossa se ei häiritse muita toimijoita. Passiivisessa tilassa kuitenkin yritetään lähettää ja vastaanottaa tarvittavat viestit. Jos virheiden muodostus jatkuu edellä mainitunkin jälkeen, tiputetaan solmu automaattisesti pois väylältä kokonaan. (Virtanen, 24.)

### 3.5 CAN-väylän diagnostiikka

Väylällä liikkuu paljon erilaisia viestejä sekä sanomia, mutta millä voidaan varmistua tietoliikenteen olevan kunnossa? Tai että väylä on edes fyysisesti toimiva? Diagnostiikkaa väylällä peruslaitteistolla on hyvin vaikea suorittaa. Tavallisella yleismittarilla ei voi kuin mitata kaapeleiden tai päätevastusten resistanssia, koska vaikka tiedettäisiin periaatteellisesti väylän jännitetasot, on bitin pituus niin lyhyt, ettei yleismittarista ole mitään hyötyä. Oskilloskoopilla saadaan liikenne väylällä näkymään kanttiaaltona, mistä pystytään päättelemään, onko väylä fyysisesti kunnossa. Mikäli kanttiaallon jännitetasot eivät vastaa väylällä käytettäviä tai ne eivät ole yhdenmukaisia toistensa kanssa, on hyvin todennäköisesti väylän kaapeleissa vikaa. Kuitenkin vaikka väylä olisi kunnossa, ei tiedetä kulkeeko siellä kaikki vaadittavat sanomat. (Virtanen, 38–39.)

CAN-sanomien havainnointiin perustyökaluna voidaan pitää tietokoneeseen liitettävää CAN-liityntää ja sen mukana saatavaa monitoriohjelmaa. Näiden avulla voidaan väylälle lähettää sanomia sekä voidaan vastaanottaa sanomia. Hieman toisenlainen työkalu on CAN-emulaattori, jolla on mahdollista esimerkiksi korvata tietty solmu. Verkon kehittäjä voi kokeilla verkon toimintaa jo kehitysvaiheessa, korvaamalla puuttuvat solmut emulaattoreilla. Toinen vaihtoehto on esimerkiksi antaa emulaattorin matkia jonkin solmun toimintaa, kun testataan toista solmua. Väylän kunnolliseen testaamiseen on tarkoitettu CAN-analysaattori. Sen toimintoihin kuuluu väyläliikenteen seurauksen ja tallentaminen, vasteaikojen ja väyläkuormituksen tilastoiminen sekä virhekehysten laskeminen. Analysaattorissa voi olla myös ominaisuus, jolla pystytään

lähettämään väylälle tahallisesti virheellinen sanoma tai vastaanottamaan sanoma virheellisesti. Analysaattori on siis kokonaisuus, joka ei varsinaisesti osallistu väylälle. Tästä huolimatta on laitteita, joissa on analysaattorin ja emulaattorin ominaisuudet yhdistettynä. Nämä laitteet yleensä toimivat tietokoneella, jossa on niille luotu ohjelmisto ja CAN-liityntä. Tietokoneelle tarkoitettujen laitteiden lisäksi on olemassa pieniä käsitestereitä, joissa yleensä on rajallinen määrä samoja ominaisuuksia kuin tietokoneella käytettävissä analysaattoreissa. Saatavilla on myös CAN-loggereita, jotka yksinkertaisesti nauhoittavat väylän liikennettä myöhempää tarkastelua varten. (Alanen & Scholliers 2004, 95.)

## 4 LIN-VÄYLÄ

### 4.1 Perusteet

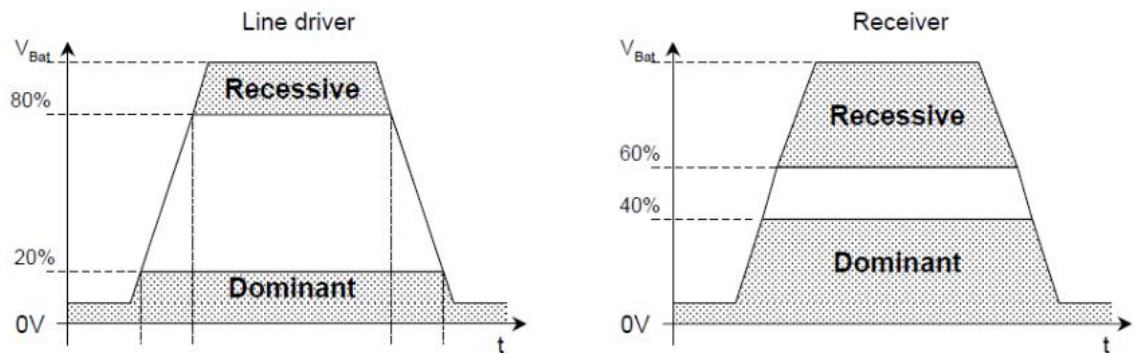
LIN eli Local interconnect Network luotiin alun perin halvaksi ja yksinkertaiseksi ratkaisuksi kohteisiin, joissa ei tarvittu CAN-väylän tasoista ratkaisua. Käyttökohteet rajoittuvat yksinkertaisuuden takia lähinnä korin alueelle, esimerkiksi ikkunoiden ohjukseen tai penkin säätämiseen. LIN-väylä perustuu yhteen hallitsevaan isäntäsolmuun ja useaan orjasolmuun. Erilaisten käyttökohteiden takia suurta nopeutta ei väylälle tarvittu ja LIN-väylän tiedonsiirron suurin nopeus onkin 20 kbit/s. Väylä on yleensä toteutettu lineaarisen topologian mukaan ja siinä käytetään vain yhtä johdinta. LIN-väylä pohjautuu aikaperustaiseen järjestelmään, jonka määrittelee isäntäsolmu. Koska kyseessä on pienelle alueelle sijoittunut paikallinen järjestelmä, on niin sanottujen orjasolmujen määrä rajattu kuuteentoista. Isäntäsolmu ohjaa orjasolmuja, jotka tyypillisesti ovat yksinkertaisia antureita, kytkimiä tai pieniä toimilaitteita. Isäntäsolmu on itse kehittyneempi ohjainlaite, joka on yhteydessä ylempään hierarkiaan eli CAN-väylään. LIN-väylä on siis CAN-väylän niin sanottu alijärjestelmä, johon se tukeutuu loppujen lopuksi. (Introduction to the LIN bus 2011; Local Interconnect Network.)

### 4.2 Tiedon välittäminen

Kuten aiemmin mainittiin, vain hallitseva isäntäsolmu hoitaa väylän viestiliikenteen aloituksen. Orjasolmuilla on mahdollisuus vastata tai toimia vain isäntäsolmun käskystä. Isäntä voi kohdistaa viestinsä yhdelle tai useammalle orjalle tai se voi sallia kahden orjan välisen tiedonsiirron. LIN-väylällä ei tarvita samanlaista törmäysten hallintaa kuin CAN-väylällä, koska pelkästään isäntä säätelee väylälle pääsyä. Myös CAN-väylän yhteydessä selitetyt väylän hallitseva ja väistytvä tila ovat hieman erilaiset. Hallitseva vastaa 0 V jännitettä ja looginen arvo on 0. Väistytvä vastaa akkujännitettä ja looginen arvo on 1. Jännitetason vaihteluiden takia vakaa tiedonsiirto on varmistettu määrittelemällä riittävät toleranssit niin hallitsevaa kuin väistytvää tilaa varten. Vastaanotossa toleranssi on suurempi kuin lähetettäessä. (Juhala 2008, 45.)



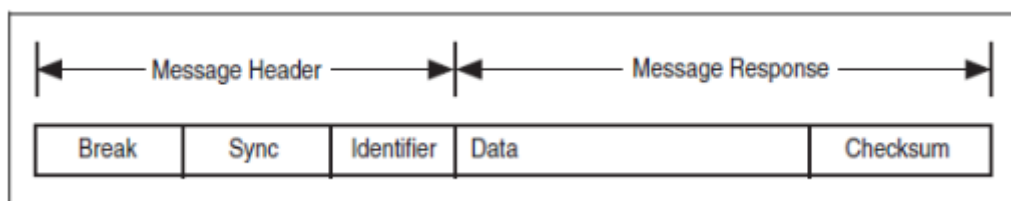
Kuvio 12 näyttää toleranssit, jossa vasemmalla lähettävän toleranssit ja oikealla vastaanottajan toleranssit (In-vehicle Networking, 24).



KUVIO 12. LIN-väylän lähettimen ja vastaanottimen toleranssit (In-Vehicle Networking, 24)

### 4.3 Viestin rakenne

LIN-väylällä liikkuva informaatio on pakattu saman tyyppiseen kehykseen, kuten aikaisemmin esitellyllä CAN-väylälläkin. LIN-väylän kehysrakenne muodostuu vain kahdesta osasta, jotka ovat viestin otsikko ja vastausosa. Kuviossa 13 on selitetty viestin eri osat (Introduction to the LIN bus 2011). Mikäli isäntäsolmu haluaa lähettää orjasolmulle käskyn suorittaa jokin tehtävä, se sisällyttää tiedon viestin vastausosaan. Jos taas isäntäsolmu pyytää jotakin tietoa, se jättää vastausosion orjan täytettäväksi, joka täydentää sen isännän haluamalla tiedolla. (Introduction to the LIN bus 2011.)



KUVIO 13. LIN-viestin kehysrakenne (Introduction to the LIN bus 2011)

Kuvioon 13 viitaten LIN-viestin kehysrakenne muodostuu seuraavista osista:

- Ensimmäisenä on tauko (Break), joka toimii viestin aloituksena. Käytetään myös nimitystä tahdistustauko.

- Toisena on itse tahdistus (Sync), jolla tahdistetaan orjien kello vastaamaan isännän kelloa.
  - Kolmantena on tunniste (Identifier), joka toimii kuten CAN-väylässä. Eli tunnisteiden perusteella solmut päättävät tarvitsevatko juuri tätä viestiä.
  - Neljäs kenttä on itse viesti (Data).
  - Viimeisenä on tarkistusluku (Checksum), jolla varmennetaan orjan vastaus.
- (Introduction to the LIN bus 2011.)

Toisessa kohdassa mainittu tahdistusmenettely mahdollistaa sen, että orjasolmujen ajalle ei määritetä tiukkoja aikavaatimuksia. Isännän kellon sen sijaan on oltava hyvin tarkka, eikä se saa poiketa kuin  $\pm 0,5 \%$  nimellisarvostaan. Orjan kello voi poiketa enintään 15 %, eikä siltikään menetetä tahdistusta. Tämä siis vain sillä varauksella, että viestin tahdistustoimenpiteet onnistuvat pienentämään poikkeaman suurimmillaan kahteen prosenttiyksikköön ennen kuin viesti päättyy. Tahdistuksen ansiosta orjasolmut pystytään toteuttamaan esimerkiksi yksinkertaisella RC-piirillä eikä tarvita kallista kvartsivärähtelijää. Kolmantena olevassa tunnistekentässä kuusi tunnisteiden kahdeksasta bitistä määrittelee itse tunnisteiden. Näillä on mahdollista muodostaa 64 erilaista tunnistetta (ID), joilla on eri merkityksensä. ID välillä 0–59 merkitsee signaalien välitystä. ID 60 tarkoittaa, että isäntä pyytää tekemään jonkun toimenpiteen tai testin ja ID 61 on taas orjan vastaus isännän viestiin. ID:t 62 ja 63 on varattuja, ensimmäinen valmistajakohtaiseen kommunikointiin ja jälkimmäinen protokolla tuleviin laajennuksiin. (Juhala 2008, 47.)

LIN-väylän kokoonpanoa eli sen toimintaa ohjaa ennalta määritetty LIN description file (ldf), johon on ohjelmoitu valmiiksi verkon toimintaa tukevia tiedostoja, esimerkiksi otsikoita ja kokoelmia. Tämä tiedosto toimii jo verkon kehittäjän apuvälineenä verkkoa luotaessa. Ldf sisältää myös ajoituslistan, jossa on määrätty kuinka usein mitäkin tietoa pyydetään. Joitakin tietoja tarvitaan tiheämmin kuin jotakin toista. Isäntäsolmu käy läpi listaa ja loppuun päästessään se aloittaa alusta, mikäli mikään muu toiminto ei sitä keskeytä. Tästä syntyy LIN-väylän deterministinen toiminta. LIN-väylän solmut voivat myös olla lepotilassa, jos väylää ei käytetä tietyn ajan sisällä tai jos isäntäsolmu määrää solmut lepotilaan. (Introduction to the LIN bus 2011.)

## 5 FLEXRAY

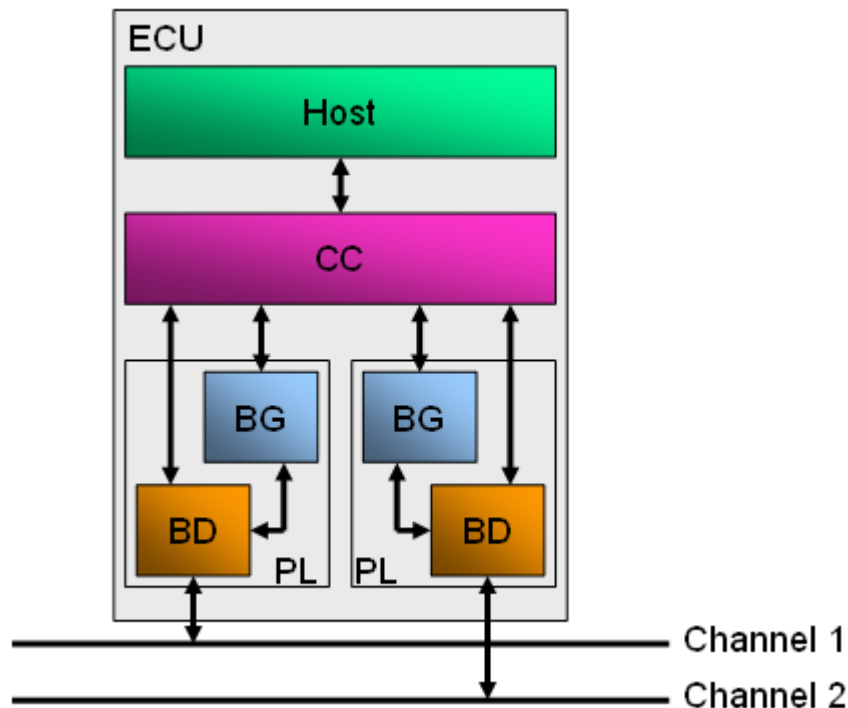
### 5.1 Perusteet

Autoteollisuuden kehittyvän turvallisuuden takia ryhdyttiin kehittämään entistä nopeampaa, vikasietoisempaa ja äärettömän luotettavaa vaihtoehtoa CAN-väylälle, joka korvaisi muut vaihtoehdot lähinnä voimalinjaan ja turvallisuuteen liittyvissä järjestelmissä. Tarkoituksena on väyläjärjestelmä, joka vastaa x-by-wire -edellytyksiä, eli poistaa mekaaniset yhteydet esimerkiksi ohjauksesta (steer-by-wire) tai jarruista (brake-by-wire) ja korvaa ne täysin sähköisellä ratkaisulla. FlexRay sai alkunsa BMW:n ja DaimlerChryslerin yhteistyöstä vuonna 1999 ja siitä on syntynyt FlexRay-yhteenliittymä, jonka jäsenenä kahden edellä mainitun lisäksi on General Motors, Motorola, Philips, Volkswagen ja Robert Bosch. Näiden lisäksi yhteenliittymällä on monia yhteistyökumppaneita. Ensimmäinen henkilöauto, jossa FlexRay on käytössä, oli BMW X5 vuonna 2006. (FlexRay Automotive Communication Bus Overview 2009; Koopman 2013, 2.)

FlexRay on väylänä luotettava, koska se tarjoaa sekä aikaperustaisen että tapahtumaperustaisen väylälle pääsyn. Kiireelliset sanomat siirtyvät aikataulun mukaisesti riippumatta muusta liikenteestä ja vähemmän kiireelliset käyttävät jäljellä jäävää kapasiteettia, jota käytetään hyvin tehokkaasti. FlexRay käyttää yleensä kierrettyä parikaapelia, mutta myös optinen kuitu on mahdollinen. Väylä on mahdollista toteuttaa kahdella välityskanavalla, jotka käyttävät eri välitysvälinettä. Tämä mahdollistaa kahdennetun tiedonvälittämisen sekä mahdollisuuden käyttää molempia kanavia suuremman nopeuden saavuttamiseksi. Väylän suurin tiedonsiirron nopeus on 10 Mbit/s, mutta käyttämällä molempia kanavia mahdollisuus on jopa 20 Mbit/s nopeuteen. FlexRay-järjestelmä on useinmiten toteutettu käyttäen lineaarista väylätopologiaa, tähtitopologiaa tai näiden yhdistelmiä. Oikea topologian valinta säästää kustannuksia. (FlexRay Automotive Communication Bus Overview 2009; Davis 2012, 2.)

## 5.2 Fyysinen kerros

FlexRay-väylän solmu on kokoonpanoltaan erilainen verrattuna CAN-väylän solmuun. Väylällä on mahdollista valita kytkeytykö solmu molempiin kanaviin vai vain toiseen. Yleensä sovelluksen määrittelemät vaatimukset ratkaisevat liittymisen tarpeen. Kuviossa 14 nähdään FlexRay-väylän solmu, joka koostuu isäntätietokoneesta (Host), tietoliikenneohjaimesta (CC) ja väyläajurista (BD) (FlexRay introduction). Väyläajureita on kummallekin kanavalle omansa. Lisäksi on mahdollista käyttää väylävahtia (BG), joita tarvitaan myös yksi kanavaa kohden. Kuten aiemmin mainittiin, FlexRay-väylällä käytetään yleisesti kierrettyä parikaapelia ja kumpikin järjestelmän kanava käsittää kaksi johdinta. Johtimet ovat väylän plus (BP, Bus Plus) ja miinus (BM, Bus Minus) -johtimet. (Juhala 2008, 86.)

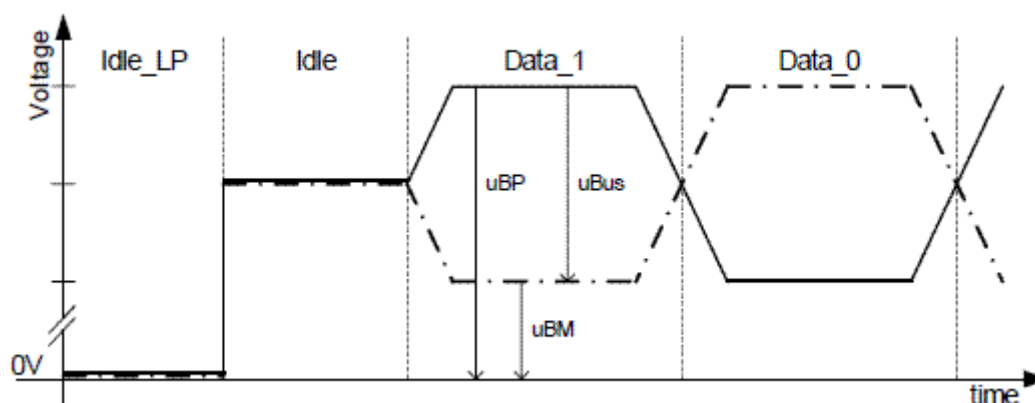


KUVIO 14. FlexRay-väylän solmu (FlexRay introduction)

FlexRay-solmussa isäntätietokoneen tehtävänä on kerätä tietoa ja välittää sitä eteenpäin tietoliikenneohjaimelle. Tietoliikenneohjain tuottaa saamastaan tiedosta bittivirran ja ohjaa väylälle pääsemistä. Lisäksi se pitää ajoituksen ja tahdistuksen samana muiden solmujen kanssa sekä luo makroaskelsignaaleita. Väyläajuri toimii lähettimenä ja se hoitaa yhteyden väylälle. Väyläajuri muuntaa loogisen informaation jännitteeksi väylälle ja toisinpäin. Sillä on myös ominaisuutena, että se pystyy valvomaan johtimien tilaa ja

osaa raportoida mahdolliset fyysiset johdinviat. Koska solmun toiminnot on mahdollista kytkeä hetkittäin pois päältä, kuuluu solmun energian hallinta väyläajurin toimenkuvaan. Väylävahtin tarkoituksena on sallia lähettäminen väylälle vain silloin, kun on kyseisen solmun vuoro lähettää. Se voi olla paikallinen, jolloin se on erikseen rakennettu jokaisen solmun sisään. Toinen vaihtoehto on keskitetty väylävahti, joka toimii yksinään keskeisellä paikalla väylässä, vahtien jokaista solmua. (Juhala 2008, 86–87.)

Kuten CAN-väylässä myös FlexRay käyttää aiemmin mainittua Non-Return to Zero menettelyä, jossa kahden peräkkäisen ja samanlaisen lähetystilan välissä ei tarvitse laskea jännitettä nollassa. FlexRay-väylässä käytettävät jännitetasot eli väylän tilat saadaan aikaiseksi käyttämällä kanavan johtimissa eri jännitteitä ( $u_{Bus}$ ) ja tila tunnustetaan mittaamalla jännite-eroa ( $u_{BP}$ - $u_{BM}$ ). Erilaisia tiloja on neljä kappaletta, jotka ovat Idle\_LP, Idle, Data\_0 ja Data\_1. Ensimmäinen tarkoittaa virransäästötilaa ja toinen tarkoittaa lepotilaa. Kaksi viimeistä merkitsevät loogista LOW-bittia ja HIGH-bittia. (FlexRay; Koopman 2013, 7.) Väylän tilat on esitetty kuviossa 15 järjestyksessä vasemmalta alkaen (FlexRay).

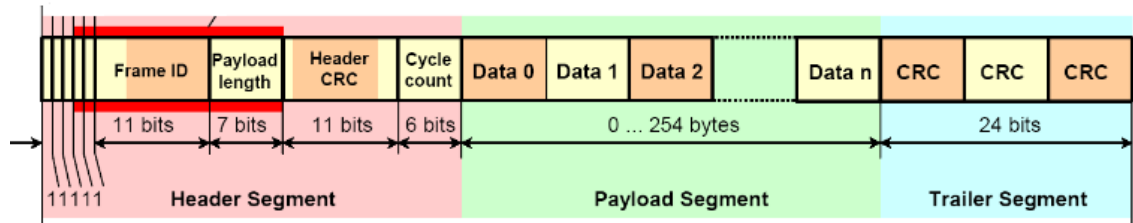


KUVIO 15. FlexRay-väylän jännitetasot ja tilat (FlexRay)

### 5.3 Siirtoyhteyskerros

FlexRay-väylä käyttää muiden ajoneuvoväylien tapaan tietynlaista viestin kehysrakennetta. Väylällä käytetään samanlaista kehystä aikaperustaisiin viesteihin sekä tapahtumaperustaisiin viesteihin. Kehysrakenne koostuu otsikosta, kuormasta ja

loppuosasta. (FlexRay Automotive Communication Bus Overview 2009.) FlexRay-viestin kehysrakenne on kuvattu kuviossa 16 (Koopman 2013, 9).



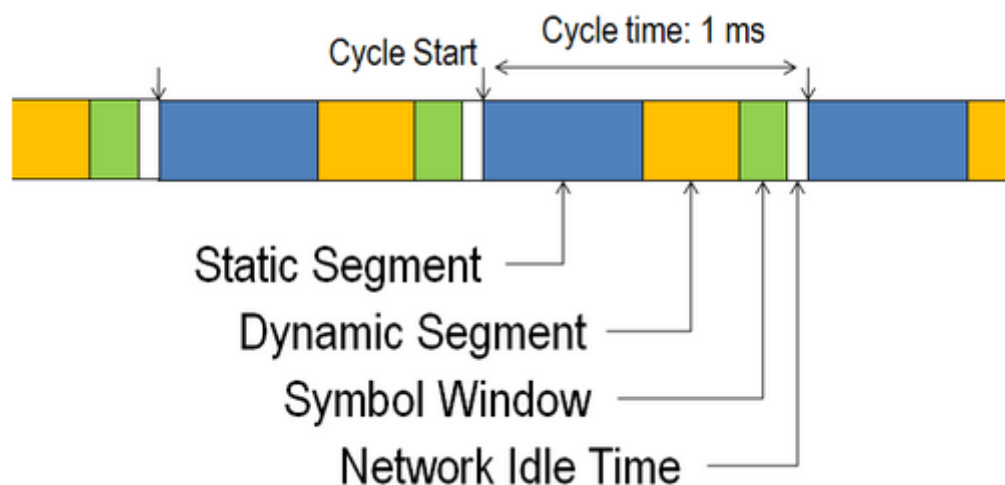
KUVIO 16. FlexRay-viestin kehysrakenne (Koopman 2013, 9)

Kuvioon 16 viitaten FlexRay-viestin kehysrakenne muodostuu seuraavista osista alkaen vasemmasta reunasta sarjalla yhden bitin tunnisteita edeten oikealle:

- Varattu bitti, joka on tarkoitettu protokollan tulevia lisäyksiä varten.
- Kuorman johdantotunniste, joka kertoo, käsittääkö kuorma verkon hallintavektorin.
- Nollakehystunniste ilmoittaa, onko kyseessä tyhjä kehys ilman käytettäväksi tarkoitettua tietoa.
- Tahdistuskehystunniste, jota käytetään järjestelmän tahdistamiseen. Vain tahdistussolmu voi asettaa tällaisen tunniste.
- Käynnistyskehystunniste, jota käytetään verkon käynnistysvaiheen aikana. Vain erityinen käynnistyssolmu voi lähettää tällaisen tunniste.
- Kehyksen tunniste (Frame ID) vastaa sen aikaikkunan numeroa, jossa kehys on lähetetty.
- Kuorman koko (Payload length) kertoo kuinka paljon tietoa lähetetään.
- Otsikon tarkistuskoodi (Header CRC).
- Syklin numero (Cycle count) ilmaisee syklin numeron ja sillä hetkellä lähetettävä solmu on aktiivinen.
- Otsikkokehyksen jälkeen tulee kuormakehys (Payload Segment), jossa ilmoitetaan itse viesti (Data). Viesti voi olla enimmillään 254 bittiä.
- Viimeisenä on loppuosa (Trailer Segment), jossa lähetetään tarkistuskoodi (CRC). (Juhala 2008, 94–95.)

FlexRay-väylällä pystytään siis lähettämään viestejä aikaperustaisesti ja tapahtumaperustaisesti. Molemmille lähetystavoille on varattu oma ikkunansa ennalta

määritellystä lähetyssaikataulusta. Puhutaan kommunikointisyklistä, joka on jaettu staattiseen ja dynaamiseen osaan. Kuvio 17 kuvaa kommunikointisykliä, jonka ensimmäinen osa on staattinen osa (FlexRay Automotive Communication Bus Overview 2009). Staattinen osa muodostuu kiinteistä lähetyssikkunoista, jotka ovat kaikki saman pituisia. Kiinteät lähetyssikkunat on jaettu verkon solmujen kesken ennalta määrättyllä jaottelulla ja näin varmistetaan, että tietyt viestit välitetään väylälle samana hetkenä joka syklistä. Staattisessa osassa solmu, jolle on määritelty oma lähetyssikkuna, pääsee siis varmasti väylälle. Toisena osana sykliä on dynaaminen osa. Dynaamisessa osassa väylälle pääsyä ohjataan miniaikaikkunoilla. Niiden määrä rajataan verkon määrittelyvaiheessa ja ne ovat kaikkien käytettävissä. Miniaikaikkunoille pääsy määrätään viestin prioriteetin mukaan. Saattaa siis olla, että vähemmän tarpeellinen tieto joutuu odottamaan monta kommunikointisykliä, päästäkseen väylälle. Sykliä jälkeen on merkki-ikkuna, joka ei ole pakollinen, mutta jolla voidaan esimerkiksi eritellä kylmäkäynnistysykli. Viimeisenä on verkon joutokäyntiaika, jossa solmuille on mahdollisuus tehdä säätöjä ja korjauksia sykliin. (FlexRay Automotive Communication Bus Overview 2009.)



KUVIO 17. FlexRay-väylän kommunikointisykli (FlexRay Automotive Communication Bus Overview 2009)

FlexRay-väylän täsmällisen kommunikoinnin takia täytyy jokaisella verkon solmulla olla sama käsitys ajasta. Yleensä hajautetuissa ohjausjärjestelmissä käytetään solmun sisäisiä kelloja, mutta ne poikkeavat nopeasti todellisesta ajasta. Tämä johtuu lämpötilan, jännitteen ja valmistuksesta johtuvien toleranssien vaihtelusta. FlexRay-väylässä on määritelty tietyt rajat sille, kuinka suuri ero ajassa saa eri solmuilla olla. Näiden asioiden vuoksi jokainen solmu automaattisesti tahdistaa itsensä tutkailemalla

muiden solmujen lähettämiä synkronointikehyksiä ja niiden ajoitusta. Synkronointikehys kuuluu kommunikointisyklin staattiseen osaan. Solmujen käsitys aikayksiköstä muodostuu tietyn hierarkian mukaan, jossa on kolme tasoa. Ensin ovat mikroaikajaksot, jotka muodostuvat suoraan solmun kellosta ja ovat voimassa vain kyseisessä solmussa. Seuraavalla tasolla on makroaikaikkunat. Niiden aikana voidaan lähettää täsmälleen yksi kehys ja ne sisältävät satunnaisen määrän mikroaikajaksoja. Ylimmällä tasolla on kommunikointisykli, joka sisältää tietyn määrän makroaikaikkunoita. Syklin makroaikaikkunoiden määrä on sama joka solmussa syklistä toiseen. Tällä tavalla solmujen käsitys ajasta säilyy muuttumattomana ja se takaa virheettömän tiedonsiirron verkossa. (FlexRay Consortium 2005, 169–171.)



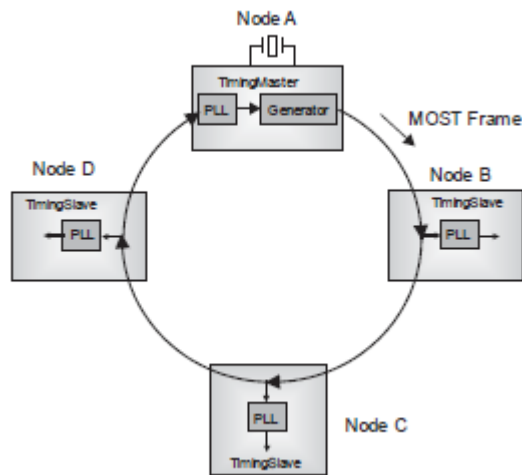
## 6 MOST-VÄYLÄ

### 6.1 Perusteet

MOST eli Media Oriented Systems Transport on väylä, joka on tarkoitettu viihde- ja informaatiokäyttöön. Nykyaikaisessa ajoneuvossa pelkästään äänen välittämisen lisäksi hyvälaatuisen liikkuvan kuvan välittäminen on kasvattanut erilaisia haasteita tekniikalle. Esimerkiksi edustusluokan autossa saattaa olla DVD-soitin, Internet-yhteys, navigointijärjestelmä ja peruutuskamera tavallisen autosoittimen lisäksi. Tämä kaikki vaatii erittäin nopeaa tiedonsiirtoväylää sekä tarkkaa tahdistusta laitteistojen välillä. Näillä vaatimuksilla MOST-väylä kehitettiin vuonna 1998 MOST-yhteistyöryhmän toimesta. Yhteistyöryhmän perustajiin kuului BMW, DaimlerChrysler, Harman/Becker ja Oasis Silicon Systems. Ryhmä tuottaa ja hallinnoi määrittelyjä perustuen MOST-väylään. Tämän lisäksi se määrittelee MOST-toteutuksille asetettavat vaatimukset ja tuottaa yhteensopivuustestausta testausyritysten kautta. Tänä päivänä MOST-yhteistyöryhmän (2014) mukaan mukana on 16 autonvalmistajaa ja alihankkijoita yli 60. (Juhala 2008, 60.)

MOST-väylän verkko on yleisimmin toteutettu rengastopologian mukaisesti niin, että jokainen solmu on yhteydessä kahteen viereiseen solmuun. Tieto väylässä liikkuu yhteen suuntaan ja jokaisella solmulla on ohitusmahdollisuus, jonka avulla solmu välittää tiedon eteenpäin, mutta ei varsinaisesti osallistu väylän toimintaan. Kuviosta 18 nähdään, että väylässä yksi solmuista toimii ajoitusisäntänä (TimingMaster), joka lähettää tietokehykset, ja muut solmut (TimingSlave) tahdistuvat tämän avulla (Grzempa 2010, 36). MOST-väyliä on kolme versiota, jotka nimetään loogisesti nopeuden mukaan. Väylät ovat MOST25, MOST50 ja MOST150, ja numerot nimen jälkeen ilmaisevat tiedonsiirtonopeuden MBit/s. Useimmiten tiedonsiirtovälineenä käytetään optista muovikuitua, mutta myös sähköinen siirto kuparikaapelin avulla sekä optinen tiedonsiirto lasikuitua tai laserdiodeja hyväksi käyttäen on mahdollista. Väylä tukee kolmea erilaista kanavaa ja niiden attribuutteja. Ohjauskanavaa käytetään ohjauskomentojen välittämiseen, tilatietojen välittämiseen ja järjestelmän kannalta tarpeellisten viestien välittämiseen. Tahdistettu kanava välittää multimediaa eli ääntä ja liikkuvaa kuvaa.

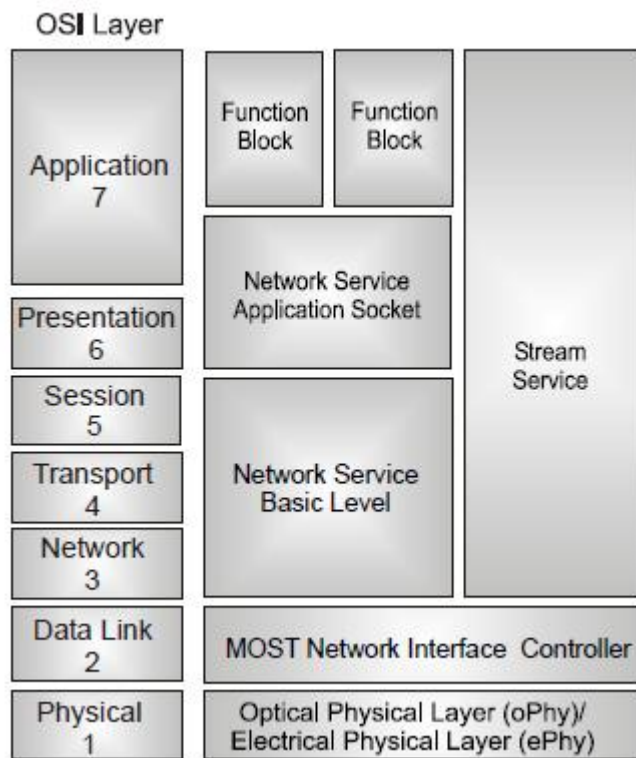
Tahdistamattomalla kanavalla tieto liikkuu paketteina ja se soveltuu toimintoihin, jotka eivät vaadi kiinteää siirtonopeutta, mutta tarvitsevat suurta siirtonopeutta. Kaistanleveyttä voidaan jakaa tahdistettujen ja tahdistamattomien kanavien kesken. (Juhala 2008, 60–61.)



KUVIO 18. MOST-väylän topologia ja solmut (Grzemba 2010, 36)

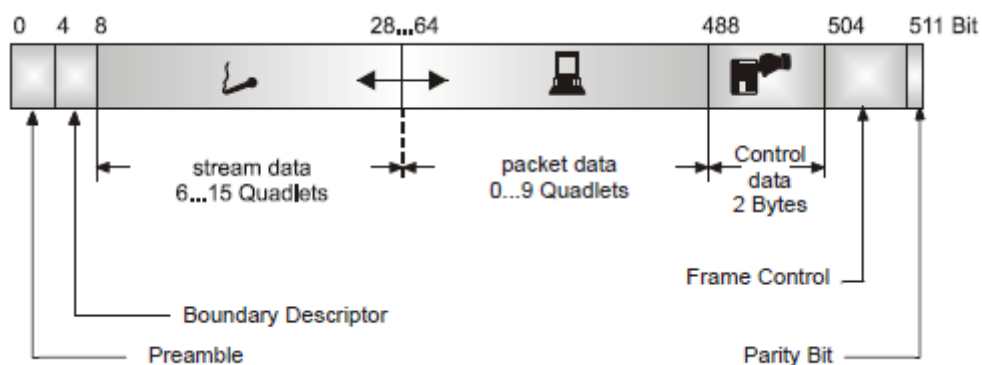
## 6.2 Laitteisto ja tiedonsiirto

MOST-väylän solmun malli on määritelty MOST-standardissa. Kuviosta 19 nähdään vaadittavat elementit ja että solmun malli toteuttaa kaikki OSI-mallin kerrokset (Grzemba 2010, 30). Kuten muissakin ajoneuvoväylissä, alimmalla tasolla fyysisessä kerroksessa toteutetaan pääsy väylälle. Seuraavalla tasolla on verkon liitynnänohjaus (Network Interface Controller), joka vastaa fyysisen kerroksen toimintojen ohjaamisesta sekä suorittaa siirron peruspalveluja. Tämän yläpuolella on verkkopalvelinkerros (Network Service), joka toimii ohjainkerroksena ja on jaettu kahteen kerrokseen. Alempi kerros (Basic Level) tarjoaa kommunikoinnin ja hallinnan peruspalvelut. Ylempi kerros (Application Socket) tarjoaa sovellusten kehittämiseen tarvittavat palvelut. Laitteen sovellutukset toteutetaan ylimmässä kerroksessa toimintalohkon (Function Block) kautta. Laitteessa saattaa olla useampia eri toimintoja ja ne voidaan näin esittää erillisinä toimintalohkoina. (Juhala 2008, 61–62.)



KUVIO 19. MOST-väylä OSI-mallissa (Grzemba 2010, 30)

Tiedon siirtäminen tapahtuu kuten muissakin ajoneuvoväylissä, tietynlaisen viestikehysen mukaan. MOST-väylässä tiedonsiirron hoitaa ajoituksesta vastaava isäntälaitte ja viesti siirtyy aina seuraavalle rengastopologian mukaisesti. Viestikehys koostuu pääosin samoista osista eri nopeuden MOST-väylissä. Kuviossa 20 nähdään MOST25-väylän viestikehys, jonka pituus on yhteensä 64 tavua (Grzemba 2010, 88).

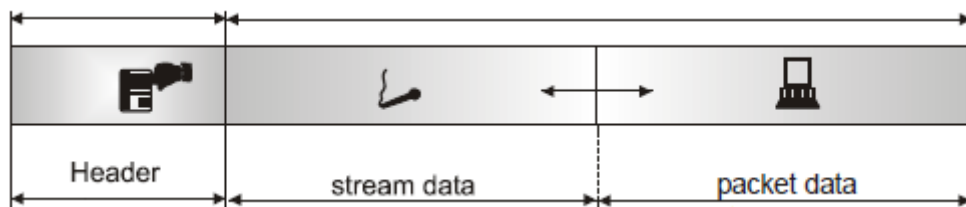


KUVIO 20. MOST25-väylän kehysrakenne (Grzemba 2010, 88 muokattu)

Viitaten kuvioon 20 MOST25-väylän kehysrakenne koostuu seuraavista osista:

- Tahdistustieto (Preamble), jolla tahdistetaan muut solmut.
- Rajakuvaus (Boundary Description), joka määrittää tahdistetun ja tahdistamattoman osuuden pituudet.
- Tahdistettujen kanavien alue (Stream Data).
- Tahdistamattomien kanavien alue eli pakettitiedon osio (Packet Data).
- Ohjauskanavan alue (Control Data).
- Kehyksen hallinta (Frame Control).
- Pariteetti (Parity Bit), virheiden tarkastelua varten. (Grzemba 2010, 88–89.)

Kuviosta 21 nähdään, että MOST50- ja MOST150-väylän viestien kehysrakenne on hyvin samankaltainen keskenään, eikä suurta eroa ole MOST25-väylän rakenteeseen (Grzemba 2010, 89). Kehysrakenteen koko kasvaa kuitenkin huomattavasti. MOST50-väylän koko on 128 tavua ja MOST150-väylän 384 tavua. Molemmissa nopeissa väylissä otsikko (Header) osuus sisältää tarvittavat tahdistukset, rajakuvaukset, ohjauskanavan alueen sekä virheen tunnistukset. Muuten loppuosa rakenteesta on samanlainen sisältäen tahdistetun ja tahdistamattoman alueen. (Grzemba 2010, 89–90.)



KUVIO 21. MOST50- ja MOST150-väylien kehysrakenne (Grzemba 2010, 89 muokattu)

## 7 ON-BOARD DIAGNOSTICS

### 7.1 Järjestelmän tarkoitus ja kehittyminen

On-Board Diagnostics eli lyhyemmin OBD on auton sisäinen valvontajärjestelmä, jonka tehtävänä on valvoa pakokaasupäästöihin vaikuttavia vikoja moottorin toiminnassa ja raportoida mahdollisista häiriöistä vikamerkkivalon avulla. Järjestelmä valvoo jatkuvasti moottorin palamistapahtumaa ja antureiden toimintaa, ja mikäli häiriötä esiintyy, syttyy vikamerkkivalo MIL (Malfunction Indicator Light) ja vikamuistiin tallentuu vikakoodi DTC (Diagnostic Trouble Code). Tänä päivänä OBD-järjestelmän vikalokia valvotaan määräaikaikatsastuksessa ja mahdolliset viat aiheuttavat hylkäyksen. (OBD-järjestelmä.)

Jo ennen virallista OBD-järjestelmää autonvalmistajilla oli käytössä yksinkertaisia järjestelmiä, jotka ilmaisivat viallisista komponenteista vilkkuvalokoodilla. Ensimmäisenä virallisen OBD-järjestelmän otti käyttöön Kalifornian osavaltio autoissa vuosimallista 1988 lähtien. Järjestelmää kehitti Kalifornian ympäristöviranomainen CARB (California Air Resources Board). Tätä järjestelmää kutsutaan nimellä OBD-I. Tekniikka kuitenkin kehittyi eikä ensimmäinen versio OBD:sta ollut standardoitu, joten seuraavan version kehitys alkoi. Vuosimallista 1994 lähtien Kalifornian osavaltiosta alkaen otettiin uusissa autoissa käyttöön OBD-II. Vuoteen 1996 mennessä koko Yhdysvalloissa järjestelmä oli pakollinen uusissa autoissa. (Overview of OBD and Regulations, 1–4.) Eurooppalainen versio EOBD pohjautuu hyvin pitkälle amerikkalaisten kehittämään järjestelmään ja se käyttää samanlaista standardoitua 16-napaista diagnoosiliitäntää. EOBD on määrätty EU:n direktiivissä ja se on pakollinen vuosimallista 2001 bensiinikäyttöisissä autoissa ja vuosimallista 2004 dieselkäyttöisissä autoissa. (OBD-järjestelmä.) Määräaikaikatsastuksessa tarkastetaan ottomoottorisen 1.1.2001 jälkeen käyttöön otetun M1- ja N1-luokan ajoneuvon OBD-järjestelmän vikaloki pakokaasupäästömittauksen yhteydessä (Trafí 2011, 1).

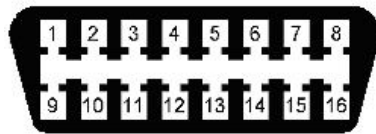
## 7.2 OBD-protokollat ja diagnoosiliitäntä

OBD-II käyttää erilaisia protokollia kommunikointiin diagnoosilaitteiston ja järjestelmän välillä. Diagnoosiliitäntä eli OBD-pistoke on standardoitu SAE:n J1962 / ISO 15031-3 standardissa. Erilaiset protokollat määrittävät, minkälaisella testilaitteella järjestelmän tietoihin pääsee käsiksi, ja eri autonvalmistajat saattavat käyttää eri protokollaa. Protokolla eroavaisuuksia oli lähinnä OBD-II-järjestelmän alkutaipaleella, koska nykyään pääsääntöisesti käytetään CAN-väylään pohjautuvaa protokollaa. Yhdysvalloissa on määrätty vuodesta 2008 lähtien, että CAN-väylää hyödyntävä protokolla on pakollinen jokaisessa autossa. Useat autonvalmistajat hyödyntävät omia protokolliaan piilottaakseen omat autoon liittyvät diagnoositietonsa, esimerkiksi laajempaan diagnosointiin tarkoitetut tiedot tai testaustoiminnot. Tämä siis tarkoittaa sitä, että yksinkertaisella testilaitteella pääsee lukemaan OBD-järjestelmään liittyvät diagnoositiedot, mutta kaikki muu autoon liittyvä diagnostiikka on saatavilla vain autonvalmistajan omalla testilaitteella. (OBD-II Protocols.)

OBD-II-järjestelmän käyttämät protokolla ovat:

- ISO 15765-4 (CAN), uusin ja modernein väylätekniikka hyödyntävä protokolla.
- ISO 14230-4 (KWP2000), K-johdinta / L-johdinta käyttävä protokolla.
- ISO 9141-2, K-johdinta / L-johdinta käyttävä protokolla.
- SAE J1850 (VPW), yleinen General Motorsin ajoneuvoissa.
- SAE J1850 (PWM), yleinen Fordin ajoneuvoissa. (OBD-II Protocols.)

Käytetyn protokollan mukaan määräytyvät myös OBD-pistokkeessa käytettävät liittimet. Kuvio 22 kertoo liittimet numeroituna ja selittää niiden merkityksen (OBD2/EOBD eri protokollat). Määrittelemättömät liitinpaikat eivät ole käytössä OBD-järjestelmässä ja näin ne ovat autonvalmistajan käytössä, mikäli niitä vain tarvitaan. Liitin on standardoitu 16-pinninen liitin ja on määritelty, että sen on sijaittava lähellä kuljettajan istuinta. Kuten aiemmin mainittiin OBD-II ja EOBD käyttävät samanlaista diagnoosipistoketta. (Muller 2005, 42–44.)

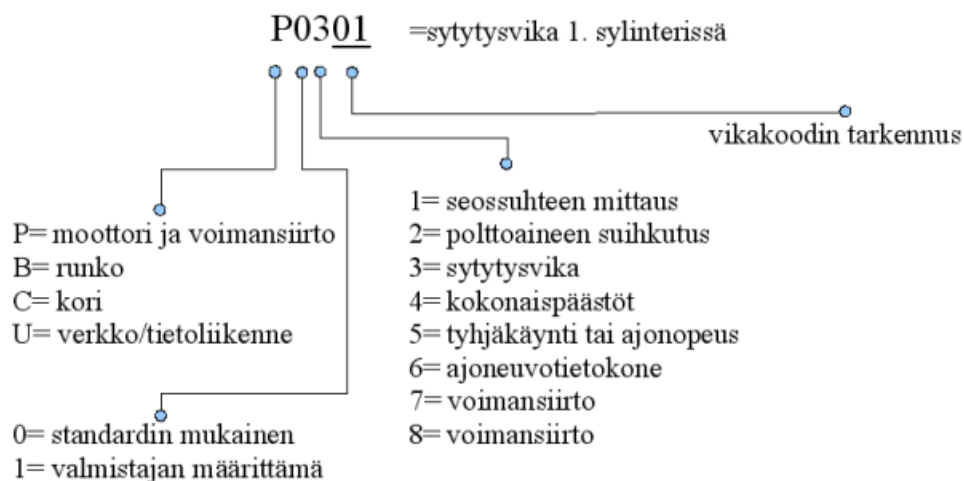


1. -
2. Positiivinen väylä SAE-J1850:ssä
3. -
4. Korin maatto
5. Signaalin maatto
6. CAN High line ISO 15765-4:ssä
7. K-linja ISO 9141-2:ssa ja ISO 14230-4:ssä
8. -
9. -
10. Negatiivinen väylä SAE-J1850:ssä
11. -
12. -
13. -
14. CAN LOW ISO 15765-4:ssä
15. L-linja ISO 9141-2:ssa ja ISO 14230-4:ssä
16. Pysyvä positiivinen jännite

KUVIO 22. OBD-II / EOBD -diagnoostipistoke (OBD2/EOBD eri protokollat)

### 7.3 Järjestelmän vikakoodit ja toiminnot

Järjestelmän vikakoodit (DTC) pyrkivät tarkentamaan viallisen komponentin havaitsemisessa. Vikakoodi koostuu 5 merkkiä pitkästä kirjaimen ja numeroiden sarjasta. OBD-II ja EOBD-järjestelmien vikakoodit ovat standardoituja, mutta samanlaisella koodilla voidaan ilmaista myös autonvalmistajan omia vikakoodeja. Autonvalmistajan omia vikakoodeja ei ole missään standardissa, joten niitä ei ole pakko edes esittää OBD-järjestelmän kautta. Kuviossa 23 on havainnoitu vikakoodin selvitykset merkki kerrallaan (OBD-järjestelmä). Vikakoodeja voi esiintyä hetkellisinä tai jatkuvina. Hetkellinen vika esiintyy vain kerran ja korjaantuu itsestään. Eli siis jokin arvo saattaa poiketa hetkellisesti sallitusta, mutta palautuu lyhyen ajan sisällä takaisin sallittuihin rajoihin. Jatkuva vika on nimensä mukaisesti pitkäkestoisempi poikkeama raja-arvoista. (OBD-Järjestelmä.)



KUVIO 23. OBD-II ja EOBD -järjestelmän vikakoodin rakenne (OBD-järjestelmä)

EOBD-standardissa määritetään järjestelmän tarjoamat testausmuodot, jotka saattavat hieman vaihdella käytetystä protokollasta ja testilaitteesta riippuen. Kuvio 24 listaa toiminnot selityksineen (Muller 2005, 42–44). Kohdassa kaksi mainittu jäädytetty kehys (Freeze Frame) tarkoittaa, että vian ilmestyessä järjestelmään ja vikakoodin tallennuttua, järjestelmä tallentaa myös tärkeimmät parametrit samalla hetkellä. Tällaisia arvoja on esimerkiksi kaasuläpän asento, moottorin kuormitus, jäähdytysnesteen lämpötila ja polttoaineen paine. (OBD-järjestelmä; Muller 2005, 42–44.)

EOBD-standardin määrittämät testausmuodot	
01	Antaa tosiaikaiset tiedot
02	Antaa ns. jäädytetyn kehyksen tiedot
03	Antaa vikakoodit
04	Poistaa vikakoodit ja tallennetut arvot
05	Antaa lambda-anturin itsetestauksen tulokset
06	Antaa itsetestauksen tulokset järjestelmistä, joita ei valvota jatkuvasti
07	Antaa itsetestauksen tulokset sellaisille järjestelmille, joita tarkkaillaan jatkuvasti
08	Erityisohjausmoodi
09	Ajoneuvodatan kuten ohjelmistoversion tai ajoneuvon tunnistenumeron (VIN) pyyntö

KUVIO 24. EOBD-standardin määrittämät testausmuodot (Muller 2005, 42–44)

Järjestelmään ilmaantuneen toimintahäiriön sattuessa syttyy mittaristoon vikamerkkivalo eli MIL. Merkkivalo on keltainen moottorin kuva, joka syttyy ja sammuu autoa käynnistettäessä, valon toiminnan testaamiseksi. Myös katsastuksessa vikamerkkivalon on toimittava, eikä se saa palaa taukoamatta. Mikäli valo palaa



taukoamatta, on järjestelmässä toimintahäiriö, mutta jos valo vilkkuu, on ajamista rajoitettava mahdollisuuksien mukaan. Tämä sen takia, koska vilkkuva valo ilmoittaa yleensä sytytyshäiriöistä, jotka päästävät palamatonta polttoaineseosta pakoputkistoon ja katalysaattoriin rikkoen sen. Toinen katsastukseen vaikuttava asia on järjestelmän valmius, josta käytetään nimitystä Readiness-koodi. Järjestelmän vikamuistin tyhjennyksen tai esimerkiksi auton akun irrotuksen jälkeen järjestelmän vikamuisti on luonnollisesti tyhjä, eikä mahdollisia toimintahäiriöitä pystytä havaitsemaan heti auton ensimmäisen käynnistyksen yhteydessä. Toimintahäiriöiden todentamiseksi autolla tarvitsee ajaa tietynlainen ajosykli, jonka aikana järjestelmä suorittaa kaikki tarvittavat testaukset ja osaa kertoa ovatko kaikki komponentit toimintakunnossa. Tällä ehkäistään myös järjestelmän tyhjentämistä ennen katsastusta ja näin mahdollisten vikojen piilottamista. Ajosykli sisältää tietyn määrän erilaisia ajotilanteita, joilla on mahdollista suorittaa järjestelmän osajärjestelmätestit. (EOBD/OBD2 määritelmät.)

## 8 TESTILAITEVERTAILU

### 8.1 Ammattimainen testilaite

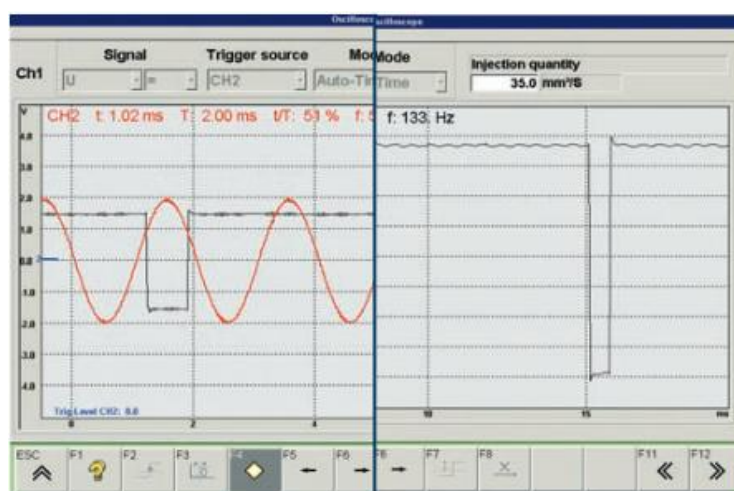
Ammattimaiseen vianhakuun autokorjaamoilla soveltuva Bosch KTS 570 edustaa testilaitteistojen huippua, silloin kun puhutaan ns. merkkiriippumattomista laitteista. Tällä tarkoitetaan, että laitetta ei ole valmistettu vain tietyn automerkin testausta varten, vaan se soveltuu usean automerkin diagnoosilaitteeksi. Robert Bosch GmbH on edelläkävijä testilaitteiden kehittämisessä, koska yritys on toiminut ajoneuvoteollisuuden alihankkijana useiden vuosikymmenien ajan. Pitkä historia alalla ja toimivat tuotteet merkitsevät, että yrityksen järjestelmiä ja laitteistoja on käytössä usean autonvalmistajan tuotteissa.

KTS 570 -järjestelmätestauslaite tarvitsee toimiakseen tietokoneen, johon asennetaan testilaitteen ohjelmisto sen käyttämiseksi. Itse testilaite, joka nähdään kuviossa 25, on lähellä taskukirjan kokoa ja se liitetään tietokoneeseen johdolla tai langattomasti (Bosch). Tämän lisäksi vaaditaan ohjelmistolisenssi, joka tavallisesti ostetaan vuodeksi kerrallaan. Lisenssi sisältää päivitykset ohjelmistoon ja näin laitteisto pysyy aina ajan tasalla. Ohjelmisto, jolla laitteistoa käytetään, on nimeltään EsiTronic ja se on räätälöitävissä monella tapaa juuri omiin tarpeisiin. Näistä eri vaihtoehdoista kerrotaan jäljempänä.



KUVIO 25. KTS 570 -järjestelmätestauslaite (Bosch)

Kuten aiemmin mainittiin, KTS 570 voidaan kytkeä tietokoneeseen joko johdolla tai käyttämällä langatonta bluetooth-tekniikkaa. Langatonta tilaa käytettäessä tietokoneeseen liitetään kuviossa 25 nähty signaalivastaanotin USB-porttiin. Kantama langattomassa tilassa on jopa 100 m. Autoon laitteen kytkentä tapahtuu OBD-pistokkeen kautta ja laite tunnistaa automaattisesti, mitä protokollaa juuri kyseinen auto käyttää. Se siis tukee kaikkia protokollia ja siihen on mahdollista saada myös adapterijohtoja, mikäli autossa ei ole standardin mukaista OBD-pistoketta. KTS 570 -laitteiston vakiovarustus on kattava ja se sisältää useita mittauksiin soveltuvia mittapäitä, koska laitetta on mahdollista käyttää yleismittarina sekä oskilloskooppina. Jännite-, virta- ja vastusmittauksia on mahdollista suorittaa kahdella kanavalla, kuten myös oskilloskooppi on kaksikanavainen. Kuviossa 26 on esitetty oskilloskoopin ominaisuuksia, joilla voidaan suorittaa tehokasta vianetsintää laittamalla komponenttien signaalit ja mitatut arvot tahdistetusti samaan kuvaan (Bosch).



#### Oskilloskoopin ominaisuudet

<b>Sisääntuloresistanssi</b>	1MΩ
<b>Kanava 1</b>	potentiaalivapaa
<b>Kanava 2</b>	potentiaalinen
<b>Näytteenottonopeus</b>	10 M näytettä / s / kanava 5 M näytettä / s / 2-kanavaa
<b>Tahdistus</b>	Useita erilaisia vaihtoehtoja
<b>Tallennus</b>	50 käyrää / 2-kanava

KUVIO 26. KTS 570 -laitteiston oskilloskoopin ominaisuuksia (Bosch)

Yleismittaria ja oskilloskooppia hallitaan siis EsiTronic ohjelmiston kautta, kuten kaikkia muitakin toimintoja laitteeseen liittyen. EsiTronicissa on muokkausvaihtoehtoja jokaisen tarpeeseen ja hintaa kertyy aina lisää, mitä enemmän ns. lisäosioita haluaa mukaan ohjelmistoon. Oletusohjelmistona pidetään EsiTronic-C-osiota, joka on

nimeltään ajoneuvodiagnoosi ja vianetsintäopas. Tällä osiolla laitetta pystyy käyttämään ja sen ympärille kootaan tarvittavat lisäosiot. Osio sisältää ohjainlaitteiden diagnosoinnin ja SIS-vianetsintäohjeet (Service Information System). Lisäksi mukaan kuuluu CAS+ (Computer Aided Service), joka automaattisesti yhdistää ohjainlaitteen diagnoosin, vianetsintäohjeen ja mitatut arvot. Yhdistämällä nämä tiedot samalle ruudulle saadaan muodostettua kokonaisvaltainen ja tehokas diagnoosi, ilman eri osioiden välillä selaamista ja etsimistä. EsiTronic-C -ohjelmiston voi valita kattamaan bensiinimoottorit, dieselmoottorit, jarrujärjestelmän tai mukavuusjärjestelmän. Valinta suoritetaan ohjainlaitediagnoosin ja SIS-vianetsintäohjeiden osioissa erikseen. Tietenkin on mahdollista valita kaikki vaihtoehdot molempiin osioihin, tai räätälöidä itselle tarpeelliset osiot ohjelmaan.


Oletusohjelmisto EsiTronic-C:n lisäksi valittavana ovat seuraavat osiot:

- EsiTronic-A ajoneuvovarusteet. Varusteiden tietokanta ja mahdollisuus vertailuun ajoneuvon alkuperäisten varaosanumeroiden kanssa.
- EsiTronic-D dieselvaraosat. Dieselkomponenttien varaosalistoja ja räjäytyskuvia.
- EsiTronic-E sähkövaraosat. Sähkökomponenttien ja varaosien tietokanta.
- EsiTronic-F varaosa-arkisto. Vanhojen ajoneuvojen varaosien tietokanta.
- EsiTronic-M ajoneuvon mekaniikka. Sisältää esim. huoltokuvat, huoltokaaviot ja tekniset ajoneuvotiedot sekä -arvot.
- EsiTronic-P sähköisten järjestelmien kytkentäkaaviot. Nimensä mukaisesti sisältää kytkentäkaaviot ja komponenttilistat sähköosista.
- EsiTronic-K komponenttien korjausohjeet diesel- ja sähköosille. Korjausohjeet, huoltotiedot ja tekniset tiedotteet otsikon mukaisille osille.
- EsiTronic-W dieseltestausarvot. Testausvaiheet dieselkomponenteille sekä mittava testiarvojen tietokanta.
- EsiTronic-B ohje- ja korjausajat. Kattava tietokanta erilaisille töille määritellyille ohje- ja korjausajoille.
- EsiTronic-S huoltovälit. Ajoneuvon valmistajan määrittämät huoltovälit ja takuun edellyttämät tarkastukset.
- EsiTronic-TD TecDoc-tiedot. Kattavat tiedot henkilö- ja hyötyajoneuvojen varaosien tunnistamiseen ja tilaamiseen.







EsiTronic-ohjelmiston käyttö on hyvin yksinkertaista ja selkeää. Käyttö aloitetaan automerkin, -mallin, vuosimallin ja moottorin valinnalla. Tämä sen vuoksi, että ohjelmisto lataa vain kyseisen auton tiedot sekä erityispiirteet ja osaa näin ollen tulkita saatuja tietoja oikein. Kuviossa 27 nähdään ohjelmiston yleiskuva. Kaikki tarvittava löytyy helposti ja otsikot kuvaavat hyvin toimintoja. Ajoneuvoinfonäkymässä on selkeästi esillä mihin järjestelmän osiin voidaan suorittaa ohjainlaitediagnoosi ja mihin löytyy vianetsintäohjeet. Diagnoosinäkymä kertoo järjestelmien tilanteen, löytyykö mahdollisia vikoja. Lisäksi siinä pystytään suorittamaan komponenttitestausta sekä laitteiden kalibrointia huolto- tai vaihtotyön jälkeen. Vianetsintänäkymästä löytyy SIS-vianetsintäohjeet, merkkikohtaiset huoltotiedotteet, tekniset huoltotiedotteet ja yleistä informaatiota erilaisista huoltotöistä. Huoltonäkymä kertoo ajoneuvon huoltovälit, tekniset tiedot ja -arvot sekä näyttää tärkeimpiä huoltokuvia. Varustusnäkymä pitää sisällään ajoneuvon varaosaluettelon ja räjäytyskuvia.

ESI[tronic]

ESI[tronic] Versio 8.0.1080 28.3.2014 11:37

 **BOSCH**

ESI[tronic] 2.0

 Ajoneuvoinfo  Diagnoosi  Vianetsintä  Huolto  Kytkeä  Varustus

Ajoneuvon yleistiedot

Järjestelmäkatsaus

Diagnoosiliitäntä

Tärkeää

Merkin liittyvää

Lyhenteet

Järjestelmä	Kuvaus	Versio	Valmistaja	SD	SIS
<b>Moottori</b>					
Ajoneuvopeus	ACC (Adaptive Cruise Control)	3.1	-	X	-
Elektron. energianhallinta	EBM-akkujärjestelmä	3.0	-	X	X
Jännitteensyöttö	Käynnistysakku	-	-	-	X
Moottorinohjaus	Motronic MED	17	-	X	X
Moottorinohjaus	Motronic MED	17.5 UDS	-	X	-
<b>Kori</b>					
Huolto	Intervallin nollaus	4.0	-	X	X
Huolto	Käsi-er. huoltovalinaytön nollaus	-	-	-	X
Huoltovalinaytö	Huoltovalinaytön nollaus KTS	-	-	-	-
Instrumentit	Mittaristo  	4.0	-	X	-
Keskuselektronikka	ZE	3.3	-	X	X
Kuljettajan info	Audio	3.0	-	X	-
Kuljettajan info	Mediasoitin	3	-	X	-
Kuljettajan info	Navigationijärjestelmä	4.0	-	X	-
Kuljettajan info	Puhelin	3.1	-	X	-
Kuljettajan info	Puhelin	4.0	-	X	-
Kuljettajan info	Puhelin/hätäkutsu	4.0	-	X	-
Kuljettajan info	Soundsystem	3.1	-	X	-
Kuljettajan info	TV-väliin	2.1	-	V	-

\* SD-diagnoosi, SIS-vianetsintä

Diagnoosi

Vianetsintä

KUVIO 27. Bosch EsiTronic 2.0 yleiskuva ohjelmistosta

Osioiden määrää tai niiden laajuutta säätelee tietenkin se, kuinka paljon sisältöä on ostovaiheessa halunnut ohjelmistoon. Tietoa, ohjeita ja sisältöä kuitenkin on mahdollista saada valtava määrä, ja yhdessä oikeaoppisen diagnosoinnin kanssa KTS 570 -laitteisto on ehdotonta kärkeiluokkaa vianetsinnässä ajoneuvoista. Yleiskorjaamoille se on todella suositeltava hankinta, mikäli on tarvetta ammattimaiseen diagnosointiin. Tavalliselle

kuluttajalle laitteisto ja sen pidempiaikainen käyttö tulee todella kalliiksi, eikä useimmissakaan tapauksissa ole hintaansa nähden kannattavaa. Atoy verkkokaupan (KTS 570 järjestelmätesteri) mukaan hinta laitteelle on 3707,60 € sis. alv. Tämän lisäksi vaadittava EsiTronic ohjainlaitediagnoosi ja korjausohjeet -ohjelmisto lisenssin kanssa maksaa 1643,00 € sis. alv. Yhteishinnaksi muodostuu 5350,60 €. Huomioonotettavaa on myös, että lisenssi on vuoden mittainen ja uusittava tämän jälkeen.

## 8.2 Kuluttajalle suunnattu testilaite

Ajoneuvojen tekniikan kehittyessä entisestään on markkinoille saapunut useita halvemman hintaluokan testilaitteita. Oikeastaan voidaan käyttää nimeä vikakoodinlukijat, koska yleisesti ominaisuuksia ei merkittävästi enempää ole. Markkinoilla olevat halvemman hintaluokan laitteet ovat tyypillisesti jonkin verran tavallista matkapuhelinta suurempia, niissä on pieni näyttö ja tarpeelliset näppäimet, ja ne liitetään johdon avulla auton diagnoosipistokkeeseen. Koska myös älypuhelimet ja tablettitietokoneet ovat kehittyneet hurjaa vauhtia ja niille tarkoitetut mobiiliapplikaatiot ovat lisääntyneet valtavasti, on testilaitteeksi tarkoitettuja sovelluksia alkanut ilmestymään. Vertailun toiseksi laitteeksi on otettu suoranaisesti kuluttajan käyttöön tarkoitettu mobiilisovellus, joka on tarkoitettu nimenomaan älypuhelimiin ja tablettitietokoneisiin.

Mitä sitten tarvitaan, jotta esimerkiksi älypuhelimesta saa tehtyä vikakoodinlukijan? Ensimmäisenä tarvitaan yhteys auton ja älypuhelimien välille. Tämä hoituu kuviossa 28 olevalla ELM327-adapterilla, joka kiinnitetään auton OBD2-pistokkeeseen (Elekma). Yhteys puhelimeen tapahtuu langattomasti bluetooth- tai Wi-Fi-yhteydellä. ELM327 adapterissa voi olla myös mahdollisuus käyttää kaapelia, jotta adapteri voidaan kytkeä vaikka kannettavaan tietokoneeseen. Adapterin lisäksi tarvitaan ohjelmisto mobiililaitteeseen, jolla saatu informaatio esitetään. Yleisimmät ohjelmistot tehdään tällä hetkellä Applen iOS tai useamman valmistajan käyttämälle Android käyttöjärjestelmille. Ohjelmistoja löytyy maksuttomia ja maksullisia, mutta hinta ei välttämättä lisää ominaisuuksia. Ohjelmisto ladataan mobiililaitteen sovelluskaupasta ja sen tilaus ei ole lisenssiluontoisesti uusittava, vaan sovellus päivityksineen on pysyvä.

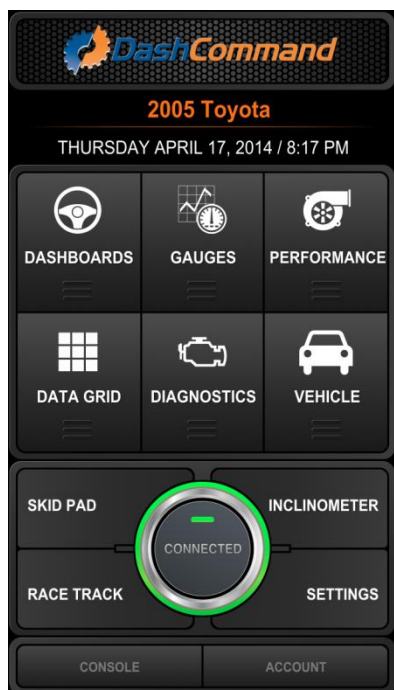




KUVIO 28. ELM327-adapteri (Elekma)

Tässä työssä käsitellään mobiilisovellusta, jonka nimi on Dashcommand. Sovellus on maksullinen, mutta kustannus käyttämisestä päivityksineen on 8,99 €. Laitteena sovelluksen testaamiseksi käytetään älypuhelinta. ELM327-adapteriksi valittiin Elekman verkkokaupasta Wi-Fi + USB malli, jossa on siis langattoman yhteyden lisäksi mahdollisuus liitää kaapelin avulla. Adapterin hinta on 49,90 €, mikä on hieman enemmän kuin pelkästään bluetooth-yhteydellä varustetun mallin hinta. Adapterin luvataan toimivan OBD-II- ja EOBD-yhteensopivissa autoissa ja tukevan kaikkia tässäkin työssä eriteltyjä OBD-protokollia. Toiminnoissa mainitaan vikakoodien luku ja poisto moottorinohjausyksiköltä sekä reaaliaikainen anturitiedon seuranta. Adapteri soveltuu iOS- ja Android-käyttöjärjestelmille sekä kuten aiemmin mainittiin, mahdollisuus myös liittää tietokoneeseen.

Dashcommand on Palmer Performance Engineering nimisen yrityksen valmistama. Yritys valmistaa OBD-diagnoosiohjelmia mobiililaitteille sekä myös tietokoneella käytettäviksi. Mobiilisovellus maksaa 8,99 € ja se on ladattavissa iOS-, Android- ja Windows-käyttöjärjestelmille. Ohjelman käyttö on hyvin selkeää ja yksinkertaista. Aluksi muodostetaan yhteys autoon valitsemalla auton merkki ja vuosimalli Vehicle-osiosta. Yhteys muodostuu pelkästään Connection-nappia painamalla, mikäli kaikki asiat ovat kohdallaan. Adapteri pitää olla toimintavalmiudessa OBD-portissa, haluttu yhteystapa avattuna, yhteysasetukset asetettu oikein ja yhteys saadaan muodostettua. Kuviossa 29 yhteys on muodostettu ja nähdään ohjelman päänäkökulma. Ensimmäisellä käyttökerralla luodaan oma tunnus, jonka avulla mahdollisesti tallennetut tiedot säilyvät samalla käyttäjällä. Tunnuksen muutokset voidaan tehdä Account-osiosta.



KUVIO 29. Dashcommand-ohjelman päänäkymä

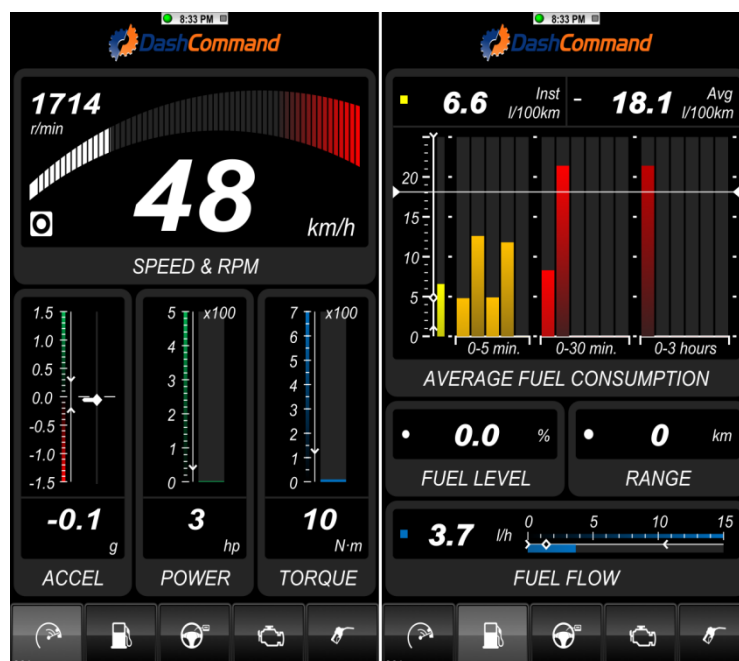
Päänäkymän kaksi ensimmäistä osiota Dashboard ja Gauges sisältävät suuren määrän erilaisia mittareita ja taulukoita, lähinnä anturitietoja sekä polttoaineen kulutukseen liittyviä tietoja. Kaikkia tietoja ei välttämättä ole saatavilla, koska kaikissa tapauksissa auton protokolla ei niitä tietoja jaa diagnoosiportin kautta tai niitä ei edes ole saatavilla. Testien perusteella todettiin, että mitä uudempi auto, sitä enemmän tietoja nämä kaksi osiota tarjoavat. Kuvioista 30 ja 31 nähdään esimerkkejä oisioiden tarjoamista tiedoista. Gauges osiossa on mahdollista muokata haluamansa mittarit ja taulukot samalle sivulle sekä muuttaa niiden kokoa. Myös Dashboard-osiossa mitta-arvoja painamalla vaihtuu mitattava suure toiseen. Seuraavassa on listattuna, mitä arvoja ohjelmisto tarjoaa:

- moottorin pyörintänopeus ja kuormitus sekä auton nopeus
- kiihtyvyysvoima (g-voima)
- imusarjan paine
- moottorin teho ja vääntömomentti
- sytytysennakko
- polttoaineen säätö lyhyellä ja pitkällä aikavälillä sekä polttoaineen paine
- imuilman, ulkoilman, katalysaattorin ja jäähdytysnesteen lämpötila
- lambda-arvo, ilman ja polttoaineen suhde sekä lambda-anturin jännite
- imuilman massa
- kaasuläpän asento.



Näiden arvojen lisäksi ohjelma näyttää arvoja polttoaineen kulutukseen liittyen. Kuten aiemmin mainittiin, kaikki arvot eivät näy jokaisessa automallissa, riippuen mitä tietoja auton ohjainyksiköt jakavat ohjelmalle. Seuraavassa esitetään lista polttoaineen kulutukseen liittyvistä arvoista:

- Polttoaineen kulutus hetkellisesti ja keskimääräisesti.
- Polttoaineen taso prosentteina ja jäljellä oleva ajomatka.
- Hiilidioksidipäästöt hetkellisesti ja keskimääräisesti.
- Kuinka pitkälle ja kuinka kauan kuljettu milläkin vaihteella.
- Kuinka pitkälle ja kuinka kauan kuljettu väärällä vaihteella.
- Polttoainekulut, mikäli polttoaineen hinta asetettu.



KUVIO 30. Dashboard-osion mitta-arvoja



KUVIO 31. Dashboard- ja Gauges-osioiden näkymät

Seuraavassa osiossa nimeltä Performance on mahdollista määrittää tietty nopeus tai matka, jonka saavuttamiseen ohjelma laskee kuluneen ajan. Tässä hyödynnetään siis älypuhelimien paikannustietoa, josta nopeus ja matka saadaan. Samaa tietoa hyödynnetään alempana olevassa Race Track-osiossa. Siinä ohjelma näyttää satelliittikartan ja piirtää ajatun reitin karttaan, merkiten eri väreillä reitin kohdat ajatun nopeuden mukaan.

Kuviossa 32 näkyvät osiot Data Grid ja Diagnostics ovat vianhaun suhteen ohjelman tärkeimmät. Ensimmäisenä mainitussa voi valita anturitiedoista haluamansa ja tallentaa niitä. Tallennuksen jälkeen tallenteen voi toistaa ja tarkkailla arvojen muutosta. Tallennus erottelee oikeaan reunaan suurimman, pienimmän ja keskiarvon mitatuista arvoista. Tallennetut tiedot on mahdollista lähettää sähköpostiin ja tutkailla niitä tietokoneella erikseen ladattavalla ohjelmalla. Diagnostics-osio on käytännössä vikakoodien lukemiseen ja poistamiseen tarkoitettu. Ohjelma ilmoittaa pysyvät ja hetkelliset vikakoodit sekä käyttäjän itsensä tallentamat. Tässä osiossa myös näkee aiemmin tässä työssä mainitun Readiness-tilan. Ohjelma ilmoittaa, onko valmiustila saavutettu, ja jos ei ole, se näyttää mitkä järjestelmätestit ovat suorittamatta.

< Menu Data Logging		< Menu Diagnostics	
707 rpm	Engine RPM	2890 rpm 0 rpm 1123 rpm	Engine Codes
0 km/h	Vehicle Speed Sensor	59 km/h 0 km/h 12 km/h	0 stored codes found 0 pending codes found 0 permanent codes found
85 °C	Engine Coolant Tempera...	85 °C 25 °C 58 °C	Readiness Monitors
22 °C	Intake Air Temperature	22 °C 15 °C 18 °C	6 complete 0 incomplete
15.5 °	Ignition Timing Advance...	31.0 ° 11.0 ° 17.0 °	
1.65 g/s	Air Flow Rate from M...	30.54 g/s 0.37 g/s 4.18 g/s	
16.9 %	Absolute Throttle Posit...	21.6 % 16.9 % 17.5 %	
15.3 %	Calculated Load Value	100.0 % 9.8 % 19.4 %	
0.605 V	Oxygen Sensor Voltag...	0.855 V 0.035 V 0.389 V	
Logging Options Add PID		Read Codes Clear Codes	

### KUVIO 32. Data Grid- ja Diagnostics-osiodien näkymä

Skid Pad ja Inclinometer kertovat enemmänkin ajaessa syntyneistä voimista ja kallistumista. Skid Pad piirtää kiihtyvyysoimia ympyränmuotoiseen asteikkoon, josta siis selviää, mihin suuntaan voimia milläkin hetkellä suuntautuu. Toiminto tallentaa voimat piirtämällä ikään kuin pisteen aina vallitsevan voiman huippuarvon kohdalle. Inclinometer eli kaltevuusmittari näyttää pystysuuntaisen ja sivuttaissuuntaisen kallistuman asteina.

## 9 POHDINTA

Väylätekniiikan kehittyminen ajoneuvoteollisuudessa on merkittävästi vienyt alan vaatimia tarpeita eteenpäin. Ongelmat, jotka ovat koituneet erilaisten järjestelmien hurjasta lisääntymisestä, on ratkaistu järkevällä ja toimivalla verkottumisella. Lisäksi elektroniikan määrän kasvaessa järjestelmän vikasietoisuus on kasvanut uskomattomaan arvoon. Uusien turvajärjestelmien kehitys lepää hyvin pitkälle innovaatioissa entistä nopeammissa ja varmemmissa väylätekniiikan ratkaisuissa. Tulevaisuudessa autoteollisuuden suuntaus tulee olemaan sähköinen yhteys yhä useampien komponenttien välillä, mekaanisen tai hydraulisen yhteyden sijasta. Työn yhtenä tarkoituksena oli koota kattava paketti nykyaikaisen ajoneuvon tiedonsiirtomenetelmistä ja luoda myös katsausta hieman vähemmän yleisessä käytössä oleviin ratkaisuihin. Tietopaketti koottiin pohjautuen kirjallisuus- ja verkkolähteisiin sillä tavalla, että kokonaisuus rakentui selkeäksi ja informatiiviseksi.

Tarkoituksena oli myös tutkia vianhakua autossa ammattilaisen testilaitteistolla ja kuluttajalle suunnatulla laitteistolla. Lopputulos on hyvin odotetunlainen. Lähtökohtana pohdinnalle on hyvä muistuttaa molempien laitteistojen hinnoista. Ammattimainen Bosch KTS 570 -järjestelmätesteri maksaa ohjelmistoineen ja lisensseineen 5350,60 € ja kuluttajalle suunnattu mobiilisovellus sekä tietojen siirtämiseen tarkoitettu adapteri maksaa 58,59 €. Hintaeroa kertyy tuhansia euroja, joten lopputuloksesta voidaan vetää jonkinlaisia johtopäätöksiä. Työn etenemisen yhteydessä aloitettiin pohtiminen, että onko halvemman hintaluokan laitteistosta minkäänlaista konkreettista hyötyä vai pelkkää hetken iloa. Suoranainen vertailu on turhaa, mutta kahden aivan eri käyttäjäkunnan laitteista pystytään toteamaan, minkälaisia ominaisuuksia tarvitaan nykypäivän auton vianhaussa. Halvemman hintaluokan laitteissa ominaisuuksia on murto-osa verrattuna kalliisiin, ja jäljelle jäävät ominaisuudetkin ovat vasta hyvä alku kunnollisessa vianhaussa. Myös kuluttajalle suunnattujen mobiilisovellusten vertaileminen nähtiin turhana, koska lähes kaikki sovellukset tarjoavat samat ominaisuudet vain hieman eri ulkoasussa. Testin aikana kylläkin kokeiltiin useampia sovelluksia, mutta Dashcommand valikoitui vaikuttavan kehityshistoriansa takia testiin, eikä näin ollen nähty tarpeelliseksi liittää työhön useamman samanlaisen sovelluksen vertailua.

Minkälaista välineistöä nykyajan auton vianhaussa sitten tarvitaan? Ensimmäinen asia, jolla vianhaku yleensä aloitetaan, on vikakoodien lukeminen. Tämä vaihe pystytään suorittamaan siis myös halvemmalla laitteistolla, mikäli vika liittyy moottorin toimintaan, eikä esimerkiksi turvavarusteiden toimintaan. Vian esiintymisestä riippuen järjestelmän muistiin tallentuu pysyvä tai hetkellinen vikakoodi, mutta mahdollista on myös, että järjestelmään ei tallennu ollenkaan vikakoodia. Jos vikakoodi on tallentunut, on ensisijaisen tärkeää ymmärtää sen merkitys ja kartoittaa mahdolliset syyt sen ilmentymiseen. Mikäli vikakoodia ei ole tallentunut, on pohdinta kohdennettava vian laatuun ja olosuhteisiin, joissa vika esiintyy. Oli vikakoodia tallentunut tai ei, useasti jatkovaiheessa tarvitaan mittauksia, toimilaitetestejä, ohjearvoja tai sähkökaavioita. On tietenkin tapauksia, joissa vian aiheuttaja saadaan selville visuaalisella tutkimisella tai yksinkertaisella kokeella, mutta näin helpolla ei välttämättä aina selvitä.

Kalliin ja halvan laitteiston ero syntyy siis heti alkuvaiheessa vianhakua. Työssä esitellyssä Bosch KTS -testilaitteessa on kaikki edellytykset edetä vianhaussa pidemmälle kuin mobiilisovelluksen käyttäjällä. Se kokoaa tarvittavan materiaalin ja soveltuu useisiin mittauksiin oskilloskoopin ja yleismittarin ansiosta. Mobiilisovelluksen käyttäjä pääsee kyllä alkuun vikakoodien tarkastamisella tai tallennetun datan tutkailulla, mutta jatkovaiheessa tarvitaan auttamattomasti lisälaitteistoa ja materiaalia. Useat mobiilisovelluksen ominaisuudet ovat lähinnä viihdykettä ja turhaa nippelitietoa, jolla ei ole vianhaun kanssa mitään tekemistä. Vääristynyt kuva halvan laitteen ominaisuuksista syntyy yleensä markkinoinnissa. Myyntiesitteessä kerrotaan kuinka helppoa on diagnosoida esimerkiksi katsastukseen vaikuttavien vikojen aiheuttaja. Lisäksi käyttäjät pohtivat perusteita autokorjaamon korkeille hinnoille vianhaussa, kun on mahdollisuus edullisesti suorittaa korjauksia omatoimisesti. Todellisuudessa korjaamon hinnasto koostuu ammattitaidosta, kunnollisesta välineistöstä sekä työlle myönnettävästä takuusta.

Lopullisena yhteenvetona halvemman hintaluokan testilaitteista jää hyvin vähän jäljelle. Vertailun vuoksi testiin valittu ammattimainen testilaitte osoittaa, mitä tarvitaan nykypäivän vianhakuun. Halvan hinnan vuoksi ei voi toivoakaan samoja ominaisuuksia kuluttajalle suunnatulta laitteelta, mutta myös laitteen markkinoinnissa täytyisi muistuttaa sen soveltumattomuudesta oikeaan vianhakuun. Eikä vian löytäminen ole pelkästään välineistä kiinni, vaan myös ammattitaito ratkaisee. Aiemmin mainittu nimitys, vikakoodinlukija kertoo hyvin pitkälle laitteen ominaisuudet, oli se sitten

mobiilisovellus tai itsenäinen laite. Loput ominaisuudet ovat enemmänkin hupi- kuin hyötykäyttöön tarkoitettuja. Fakta on kuitenkin se, että nykypäivän autot kehittyvät ja monimutkaistuvat tuoden haasteita jo korjaamoillekin. Silti näiden laitteiden valmistajat yrittävät tuoda vianhakua edelleen lähemmäs kuluttajaa, vaikka kuluttaja ei välttämättä ymmärrä tätä kasvavaa kuilua laitteiden markkinoinnin ja oikean vianhaun välillä.

## LÄHTEET

Alanen, J. & Scholliers, J. 2004. EN 13149 –standardin sovellusohje liite 1. Internetjulkaisu. Julkaistu 8.1.2004. Luettu 19.3.2014.  
[http://virtual.vtt.fi/virtual/proj6/fits/julkaisut/hanke1/fits33\\_2004.pdf](http://virtual.vtt.fi/virtual/proj6/fits/julkaisut/hanke1/fits33_2004.pdf)

Atoy. Bosch KTS570 järjestelmätesteri. Verkkokauppa. Luettu 16.4.2014.  
[http://www.atoy.fi/verkkokauppa/bosch\\_kts570\\_jarjestelmatesteri-p-127941-77/](http://www.atoy.fi/verkkokauppa/bosch_kts570_jarjestelmatesteri-p-127941-77/)

Bosch. KTS 570 tuotetietoa. Internetartikkeli. Julkaisuaikankohta tuntematon. Luettu 15.4.2014. [http://rb-aa.bosch.com/boaa-fi/Product.jsp?prod\\_id=326&ccat\\_id=128&language=fi-FI&publication=3](http://rb-aa.bosch.com/boaa-fi/Product.jsp?prod_id=326&ccat_id=128&language=fi-FI&publication=3)

CAN in Automation. CAN history. Internetartikkeli. Julkaisuaikankohta tuntematon. Luettu 19.3.2014. <http://www.can-cia.de/index.php?id=161>

Davis, G. 2012. Debugging automotive serial buses: CAN, LIN and FlexRay exposed. Internetartikkeli. Julkaistu 26.6.2012. Luettu 26.3.2014.  
<http://www.edn.com/design/test-and-measurement/4391269/1/Debugging-automotive-serial-buses--CAN--LIN-and-FlexRay-exposed>

Eberspächer. FlexRay introduction. Internetartikkeli. Julkaisuaikankohta tuntematon. Luettu 26.3.2014. <http://www.eberspaecher-electronics.com/en/company/flexray-introduction.html#c62705>

Electronics Engineering Herald. CAN interface in embedded systems. Internetkurssi. Julkaisuaikankohta tuntematon. Luettu 21.3.2014.  
<http://www.eeherald.com/section/design-guide/esmod9.html>

Electronics Engineering Herald. Local Interconnect Network. Internetkurssi. Julkaisuaikankohta tuntematon. Luettu 24.3.2014.  
<http://www.eeherald.com/section/design-guide/esmod10.html>

Elekma. ELM327 adapteri (bluetooth). Verkkokauppa. Luettu 16.4.2014.  
[http://www.elekma.com/elm327\\_adapteri\\_bluetooth](http://www.elekma.com/elm327_adapteri_bluetooth)

Elekma. EOBD/OBD2 eri protokollat. Internetartikkeli. Julkaisuaikankohta tuntematon. Luettu 10.4.2014. [http://www.elekma.com/eri\\_protokollat](http://www.elekma.com/eri_protokollat)

Elekma. EOBD/OBD2 määritelmät. Internetartikkeli. Julkaisuaikankohta tuntematon. Luettu 11.4.2014. [http://www.elekma.com/obd2\\_maaritelmät](http://www.elekma.com/obd2_maaritelmät)

FlexRay Consortium. 2005. FlexRay Communications System Protocol Specification Version 2.1. Revision A. Julkaistu 22.12.2005. Luettu 7.4.2014.  
[http://www.softwareresearch.net/fileadmin/src/docs/teaching/SS08/PS\\_VS/FlexRayCommunicationSystem.pdf](http://www.softwareresearch.net/fileadmin/src/docs/teaching/SS08/PS_VS/FlexRayCommunicationSystem.pdf)

Grzemba, A. 2010. MOST The Automotive Multimedia Network. Internetjulkaisu kirjasta. Julkaistu 7/2010. Luettu 8.4.2014.  
<http://www.mostcooperation.com/en/publications/most-book/>

In-Vehicle Networking. Internetartikkeli. Julkaisuaikajankohta tuntematon. Luettu 25.3.2014. <http://www.vr.ncue.edu.tw/esa/a1014/LIN.pdf>

Robert Bosch GmbH. 2008. Ajoneuvojen verkottuminen. Suom. Juhala, M. 1. painos Helsinki: Autoalan Koulutuskeskus Oy. Alkuperäinen teos 2007.

Koivisto, J., Mikkolainen, P. & Rantala, J. 2012. Autotekniikka 5. Autosähkötekniikan perusteet. 1. painos. Helsinki: Kustannusosakeyhtiö Otava.

Koopman, P. 2013. The Flexray Protocol. Internetartikkeli. Julkaistu 25.11.2013. Luettu 26.3.2014. [https://www.ece.cmu.edu/~ece649/lectures/23\\_flexray.pdf](https://www.ece.cmu.edu/~ece649/lectures/23_flexray.pdf)

McQuerry S. 2008. Chapter 1: Building a Simple Network. Internetartikkeli. Julkaistu 29.5.2008. Luettu 3.3.2014.  
<http://www.networkworld.com/redesign08/subnets/cisco/053008-ch1-ccna-prep-library.html?page=1>

MOST Cooperation. 2014. Real Interconnectivity – Network Intelligence. Yhteistyöryhmän Internet-sivusto. Päivitetty 2014. Luettu 7.4.2014.  
<http://www.mostcooperation.com/en/>

Muller, G. 2005. Diagnostiikkaa EOBD-liitännällä. Prosessori 9/2005, 42–44.

Myllymäki, R. Tuntiopettaja. 2013. Väylätekniikka. Oppimateriaali. 5.3.2014. Tampereen ammattikorkeakoulu.

National Instruments. 2009. Flexray Automotive Communication Bus Overview. Internetartikkeli. Julkaistu 21.8.2009. Luettu 26.3.2014. <http://www.ni.com/white-paper/3352/en/>

National Instruments. 2011. Introduction to the Local Interconnect Network Bus. Internetartikkeli. Julkaistu 3.11.2011. Luettu 24.3.2014.  
<http://sine.ni.com/np/app/main/p/ap/icommm/lang/en/pg/1/sn/n17:icommm,n21:9536/fmid/2954/>

OBDTester. OBD-II Protocols. Internetartikkeli. Julkaisuaikajankohta tuntematon. Luettu 10.4.2014. [http://www.obdtester.com/obd2\\_protocols](http://www.obdtester.com/obd2_protocols)

Rey, S. 2003. Introduction to LIN. Internetartikkeli. Julkaistu 13.5.2003. Luettu 24.3.2014. <http://www.scribd.com/doc/2903052/Introduction-to-LIN-Local-Interconnect-Network>

Shandle, J. 2003. CAN: Network for Thousands of Applications Outside Automotive. Internetartikkeli. Julkaistu 26.6.2003. Luettu 20.3.2014.  
[http://www.eetimes.com/document.asp?doc\\_id=1275775](http://www.eetimes.com/document.asp?doc_id=1275775)

Sixtek. OBD-järjestelmä. Internetartikkeli. Julkaisuaikajankohta tuntematon. Luettu 9.4.2014. <http://www.sixtek.fi/obd-teksti.htm>

Spancion. FlexRay. Julkaisuaikajankohta tuntematon. Luettu 31.3.2014.  
[http://www.spancion.com/Support/microcontrollers/developmentenvironment/Pages/flexray-index\\_p9.aspx](http://www.spancion.com/Support/microcontrollers/developmentenvironment/Pages/flexray-index_p9.aspx)



Toyota Motor Sales USA. Overview of OBD and Regulations. Internetartikkeli.  
Julkaisuajankohta tuntematon. Luettu 9.4.2014.  
<http://www.autoshop101.com/forms/h46.pdf>

Trafi. 2011. Ottomoottorikäyttöisten ajoneuvojen pakokaasupäästöjen tarkastus.  
Internetjulkaisu. Julkaistu 25.3.2011. Luettu 9.4.2014.  
[http://www.trafi.fi/filebank/a/1325147177/3fb9d1c954c8aab89c3d40b8cce5ca26/4756-Ottomoottorikayttoisten\\_pakokaasupaastojen\\_tarkastus.pdf](http://www.trafi.fi/filebank/a/1325147177/3fb9d1c954c8aab89c3d40b8cce5ca26/4756-Ottomoottorikayttoisten_pakokaasupaastojen_tarkastus.pdf)

Virtanen, A. CAN Ajoneuvojen ja koneiden sisäinen paikallisväylä. Opetusmateriaali  
Aalto-yliopisto. Julkaisuajankohta tuntematon. Luettu 18.3.2014.  
[www.noppa.aalto.fi/noppa/kurssi/as-116.2120/luennot/](http://www.noppa.aalto.fi/noppa/kurssi/as-116.2120/luennot/)